DevOps
INSTITUTE

DevSecOps Foundation® Exam Study Guide

DevOps Institute

DevOps Institute is dedicated to advancing the human elements of DevOps success. We fulfill our mission through our SKIL framework of Skills, Knowledge, Ideas and Learning.

Certification is one means of showcasing your skills. While we strongly support formal training as the best learning experience and method for certification preparation, DevOps Institute also recognizes that humans learn in different ways from different resources and experiences. As the defacto certification body for DevOps, DevOps Institute has now removed the barrier to certification by removing formal training prerequisites and opening our testing program to anyone who believes that they have the topical knowledge and experience to pass one or more of our certification exams.

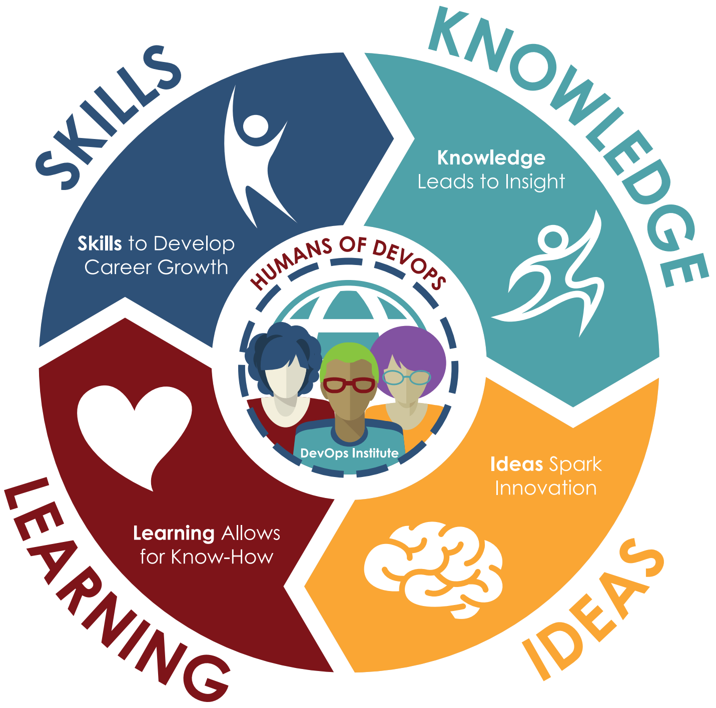
This examination study guide will help test-takers prepare by defining the scope of the exam and includes the following:

- Course Description
- Examination Requirements
- DevOps Glossary of Terms
- Value Added Resources
- Sample Exam(s) with Answer Key

These assets provide a guideline for the topics, concepts, vocabulary and definitions that the exam candidate is expected to know and understand in order to pass the exam. The knowledge itself will need to be gained on its own or through training by one of our Global Education Partners.

Test-takers who successfully pass the exam will also receive a certificate and digital badge from DevOps Institute, acknowledging their achievement, that can be shared with their professional online networks.

If you have any questions, please contact our DevOps Institute Customer Service team at CustomerService@DevOpsInstitute.com.


SKILLS
KNOWLEDGE
IDEAS
LEARNING
Skills to Develop Career Growth
Knowledge Leads to Insight
Ideas Spark Innovation
Learning Allows for Know-How
HUMANS OF DEVOPS
DevOps Institute

DevOps Institute's SKIL Framework

DevOps Institute is dedicated to advancing the human elements of DevOps success through our human-centered SKIL framework of Skills, Knowledge, Ideas and Learning.

We develop, accredit and orchestrate SKIL through certifications, research, learning opportunities, events and community connections.

Visit the SKILup℠ CAFE at www.devopsinstitute.com to learn more.

Join Us!

Become a member and join the fastest growing global community of DevOps practitioners and professionals.

The DevOps Institute continuous learning community is your go-to hub for all things DevOps, so get ready to learn, engage, and inspire.

Visit https: www.devopsinstitute.com/become-a-community-member to join today.

You belong.

DevSecOps Foundation (DSOF)℠ Course Description

DURATION - 16 Hours

Learn the purpose, benefits, concepts, and vocabulary of DevSecOps including DevOps security strategies and business benefits.

OVERVIEW

As companies deploy code faster and more often than ever, new vulnerabilities are also accelerating. When the boss says, “Do more with less”, DevOps practices adds business and security value as an integral, strategic component. Delivering development, security, and operations at the speed of business should be an essential component for any modern enterprise.

Course topics covered include how DevSecOps provides business value, enhancing your business opportunities, and improving corporate value. The core DevSecOps principles taught can support an organizational transformation, increase productivity, reduce risk, and optimize resource usage.

This course explains how DevOps security practices differ from other approaches then delivers the education needed to apply changes to your organization. Participants learn the purpose, benefits, concepts, vocabulary and applications of DevSecOps. Most importantly, students learn how DevSecOps roles fit with a DevOps culture and organization. At the course’s end, participants will understand “security as code” to make security and compliance value consumable as a service.

No course would be complete without practical application and this course teaches the steps to integrate security programs from the developers and operators through the business C-level. Every stakeholder plays a part and the learning material highlights how professionals can use these tools as the primary means of protecting the organization and customer through multiple case studies, video presentations, discussion options, and exercise material to maximize learning value. These real-life scenarios create tangible takeaways participants can leverage upon their return to the home office.

This course positions learners to pass the DevSecOps Foundation exam.

COURSE OBJECTIVES

The learning objectives include a practical understanding of:

- The purpose, benefits, concepts, and vocabulary of DevSecOps
- How DevOps security practices differ from other security approaches
- Business-driven security strategies and Best Practices

- Understanding and applying data and security sciences
- Integrating corporate stakeholders into DevSecOps Practices
- Enhancing communication between Dev, Sec, and Ops teams
- How DevSecOps roles fit with a DevOps culture and organization

AUDIENCE

The target audience for the DevSecOps Foundation course are professionals including:

- Anyone involved or interested in learning about DevSecOps strategies and automation
- Anyone involved in Continuous Delivery toolchain architectures
- Compliance Team
- Business managers
- Delivery Staff
- DevOps Engineers
- IT Managers
- IT Security Professionals, Practitioners, and Managers
- Maintenance and support staff
- Managed Service Providers
- Project & Product Managers
- Quality Assurance Teams
- Release Managers
- Scrum Masters
- Site Reliability Engineers
- Software Engineers
- Testers

LEARNER MATERIALS

- Digital Learner Manual (excellent post-class reference)
- Participation in exercises designed to apply concepts
- Sample documents, templates, tools and techniques
- Access to additional sources of information and communities

PREREQUISITES

Participants should have baseline knowledge and understanding of common DevOps definitions and principles.

CERTIFICATION EXAM

Successfully passing (65%) the 60-minute examination, consisting of 40 multiple-choice questions, leads to the candidate’s designation as DevSecOps Foundation (DSOF) certified. The certification is governed and maintained by DevOps Institute.

COURSE OUTLINE

- Realizing DevSecOps Outcomes
  - Origins of DevOps
  - Evolution of DevSecOps
  - CALMS
  - The Three Ways
- Defining the Cyberthreat Landscape
  - What is the Cyber Threat Landscape?
  - What is the threat?
  - What do we protect from?
  - What do we protect, and why?
  - How do I talk to security?
- Building a Responsive DevSecOps Model
  - Demonstrate Model
  - Technical, business and human outcomes
  - What's being measured?
  - Gating and thresholding
- Integrating DevSecOps Stakeholders
  - The DevSecOps State of Mind
  - The DevSecOps Stakeholders
  - What's at stake for who?
  - Participating in the DevSecOps model
- Establishing DevSecOps Best Practices
  - Start where you are
  - Integrating people, process and technology and governance
  - DevSecOps operating model
  - Communication practices and boundaries
  - Focusing on outcomes
- Best Practices to get Started
  - The Three Ways
  - Identifying target states
  - Value stream-centric thinking
- DevOps Pipelines and Continuous Compliance
  - The goal of a DevOps pipeline
  - Why continuous compliance is important
  - Archetypes and reference architectures
  - Coordinating DevOps Pipeline construction
  - DevSecOps tool categories, types and examples

- Learning Using Outcomes
    - Security Training Options
    - Training as Policy
    - Experiential Learning
    - Cross-Skilling
    - The DevSecOps Collective Body of Knowledge
    - Preparing for the DevSecOps Foundation certification exam

DevOps Institute

DevSecOps Foundation℠

Examination Requirements

DevSecOps Foundation (DSOF)℠ Certification

DevSecOps Foundation is a certification that is accredited by DevOps Institute. A DevSecOps engineer is an IT security professional who is skilled at security as code with the intent of making security and compliance consumable as a service. A DevSecOps engineer uses data and security science as their primary means of protecting the organization and customer. The purpose of this certification and its associated course is to impart, test and validate knowledge of DevSecOps vocabulary, principles, practices, automation and value.

Eligibility for Examination

Although there are no formal prerequisites for the exam, DevOps Institute highly recommends the following to prepare candidates for the exam leading to the DevSecOps Foundation certification:

- It is recommended that candidates complete at least 16 contact hours (instruction and labs) as part of a formal, approved training course delivered by an accredited Education Partner of DevOps Institute

Examination Administration

The DevSecOps Foundation examination is accredited, managed and administered under the strict protocols and standards of DevOps Institute.

Level of Difficulty

The DevSecOps Foundation certification uses the Bloom Taxonomy of Educational Objectives in the construction of both the learning content and the examination.

- The DevSecOps Foundation exam contains Bloom 1 questions that test learners' ***knowledge*** of DevOps concepts and vocabulary terms
- The exam also contains Bloom 2 questions that test learners' ***comprehension*** of these concepts in context

Format of the Examination

Candidates must achieve a passing score to gain the DevSecOps Foundation Certification.

Exam Type	40 multiple choice questions
Duration	60 minutes
Prerequisites	It is recommended that candidates complete the DevSecOps Foundation course from an accredited DevOps Institute Education Partner
Supervised	No
Open Book	Yes
Passing Score	65%
Delivery	Web-based
Badge	DevSecOps Foundation

Exam Topic Areas and Question Weighting

The DevSecOps Foundation exam requires knowledge of the topic areas specified below:

Topic Area	Description	Max Questions
DSOF – 1	Realizing DevSecOps Outcomes	5
DSOF – 2	Defining the Cyber Threat Landscape	6
DSOF – 3	Building a Responsive DevSecOps Model	3
DSOF – 4	Integrating DevSecOps Stakeholders	5
DSOF – 5	Establishing DevSecOps Practices	6
DSOF – 6	Best Practices to Get Started	7
DSOF – 7	DevOps Pipelines and Continuous Compliance	5
DSOF – 8	Learning Using Outcomes	3

Concept and Terminology List

The candidate is expected to understand, comprehend and apply the following DevOps concepts and vocabulary at a Blooms Level 1 and 2.

- Agile
- API
- Architecture
- Artifact Management
- Authentication
- Authorization
- Access management
- Advice process
- Binary Instrumentation
- Business Continuity Plan
- Business transformation
- CALMS
- CIA triad
- CICD Pipeline
- Chaos Engineering
- Container Security
- Continuous Compliance
- Continuous Security
- Cross-skilling
- Cyberthreat Landscape
- DevSecOps
- DIE
- DREAD
- Dynamic Application Security Testing (DAST)
- Cooperation
- Erickson
- Fuzzing
- Governance, risk management and compliance (GRC) platform
- Identity
- Identity and access management (IAM)
- Incident response
- Interactive Application Security Testing (IAST)
- Issue management
- Laloux
- Log management
- Mean Time to Change (MTTC)
- Mean Time to Detect (MTTD)
- Mean Time to Recover (MTTR)
- Multi-factor authentication
- OCTAVE
- Ops management
- Patch
- Patch management
- Penetration testing
- Policy as code
- Privileged access management
- RACI
- Real Time Application Self-Protection (RASP)
- Resilience
- Retrospective
- Risk Management
- Roles
- Role-based access control
- Safety Culture
- Secrets Management
- Security as code
- Security Information and Event Management (SIEM)
- Separation of Duties (SOD)
- Shared vision and objectives
- Shift left
- Site Reliability Engineering (SRE)
- Software Composition Analysis (SCA)
- Stakeholder Modeling
- Static Application Security Testing (SAST)
- STRIDE
- System of record
- Supply Chain
- Telemetry
- The Three Ways
- Threat
- Threat modeling
- Threat intelligence
- Value Stream
- Vulnerability
- Vulnerability management
- Vulnerability scans
- Westrum

DevOps Institute

DevSecOps Foundation

Sample Examination 1

1. Which BEST represents the goal of DevSecOps?
    a. Meet governance, risk and compliance requirements
    b. Safely distribute security decisions at speed and scale
    c. Automate security policies and audit requirements
    d. Embed security practices into software development

2. According to Laloux's advice process, what must be done by any person making a decision?
    a. Seek advice from an expert
    b. Seek advice from people who will be impacted
    c. Consider the cost
    d. Both A and B

3. Which is needed to build meaningful metrics?
    a. Data
    b. A repeatable approach
    c. Context
    d. All of the above

4. Which type of tool can be used to limit access to production by automation, orchestration and configuration management tools?
    a. Password management tools
    b. Configuration management tools
    c. Privileged access management tools
    d. GRC tools

5. If Bob Berker establishes a pipeline to deploy software in a fast and continuous manner, which of the following DevSecOps goals could he be trying to achieve?
    a. Bake security in rather than bolt it on
    b. The Third Way
    c. Quality in checks
    d. Rapid time to market

6. Which represents the BEST practices to building KPIs which reflect a responsive DevSecOps Pipeline?
    a. Whitelisting only approved applications and reporting results
    b. KPIs are driven by pipeline/application with the ability to threshold and gate at every stage
    c. Allow teams to find their own solutions
    d. Focus on meeting the audit team's information and reporting requirements

7. Which term represents the capability of an environment or organization to tolerate change and disturbances?
    a. Resilience
    b. Flexibility
    c. Agility
    d. Adaptability

8. Which approach can be used to reduce tensions between an organization's security, development and operations teams that are caused by a comprehensive security program?
    a. Introduce patterns over time so people become used to working in a certain way
    b. Set strict gates between dev and ops functions
    c. Distribute all security tasks which cause conflict to non-security individuals
    d. Implement a Governance, Risk Management and Compliance (GRC) platform

9. Jacqueline establishes a pipeline to expedite her software development and is determined to implement code-driven, peer-reviewed processes? Which of the following is she attempting to implement?
    a. Shifting security left
    b. Data Standards
    c. Data Validation
    d. Reducing technical debt

10. What is Governance, Risk Management and Compliance (GRC)?
    a. A class of tools/platforms
    b. A team or practice/program area
    c. An executive-level committee
    d. Either A or B

11. Which statement about continuous security practices is MOST correct?
    a. Represents the addressing of security concerns and testing in the Continuous Delivery pipeline
    b. Software development practice where team members integrate daily
    c. Should be fully automated
    d. Allows for every change to be processed through a pipeline and put into production

12. In the context of DevSecOps, which is an example of the 'shift left' principle?
    a. Involve security during application design
    b. Automate penetration tests
    c. Introduce threat modeling
    d. Introduce test-driven development

13. Which characteristic of resilient organizations makes it possible for them to overcome failure? The ability to…
    a. Recover quickly
    b. Prevent impact
    c. Learn fast
    d. Both A and C

14. Planning for a DevSecOps pipeline requires managing the structure through carefully implementing tools for notification, health and architectures? What type of asset categories are typically associated with architecture?
    a. Virtual Machines, Containers, Platform as a Service
    b. Infrastructure as code, Identity as a service, Apache
    c. Kafka, Kubernetes, Docker
    d. Role-based access control, Supply chain metrics, scrum

15. Which describes the purpose of dynamic application security testing (DAST) tools?
    a. Performs vulnerability and weakness analysis on source code
    b. Performs vulnerability and weakness analysis on compiled (built) code
    c. Checks for libraries or functions that have known vulnerabilities
    d. Looks for security weaknesses by gaining access to a system's data

16. Which is a trigger for the incident response process?
    a. Log data
    b. Threat intelligence
    c. Attack response data
    d. Both A and B

17. As part of a DevOps experiment, a development team has set up a test environment using cloud services. The team wants to use best practices to secure the environment. Which is NOT an IAM best practice?
    a. Store root access keys in a vault
    b. Assign permissions directly to users
    c. Rotate secrets on a cadence
    d. Enable MFA authentication for privileged users

18. The term Safety Culture most likely refers to which of the following statements?
    a. I feel free to tell my boss bad news
    b. All OSHA standards are met and information sheets posted
    c. Any incident is investigated and individuals found to be responsible are removed from the company
    d. Incidents are attributed to individuals rather than any breakdown in organizational policy

19. In your responsive DevSecOps pipeline, which elements should best be used to create a backlog for new work?
    a. Suggestions from your customers
    b. Recommended solutions from the C-Suite
    c. New vulnerabilities identified by threat intelligence
    d. Integration and Output gaps

20. An organization in a heavily-regulated industry is under tremendous pressure to bring its products to market more quickly. Software developers indicate road blocks put in place by Security Management to minimize risks are negatively impacting their ability to quickly release production-ready code. Which DevSecOps principle should this organization consider FIRST to improve its performance?
    a. Invest in security education and awareness
    b. Automate a minimum set of security practices
    c. Create a shared vision and objectives
    d. Measure for desired outcomes

21. An organization has a *very limited* budget. A team is investigating ways to improve application security testing. Which testing technique will BEST meet their current needs?
    a. Static application security testing
    b. Dynamic application security testing
    c. Software composition analysis
    d. Penetration testing

22. A development team wants to replicate full original production data to conduct a series of tests. In the context of DevSecOps Engineering, what conditions must be met for this to happen?
    a. Production data should never be used for testing
    b. Backup the data prior to testing to mitigate risks
    c. Store the data in a fully production-secure environment
    d. Mask sensitive data after it is replicated

23. Which testing type compliments an organization's continuous integration practices?
    a. Penetration tests
    b. Vulnerability scans
    c. Canary tests
    d. Static application security tests

24. An organization's DevOps efforts have stalled due to audit concerns. Which DevSecOps practices can help alleviate audit's concerns?
    a. Map changes to approved users and change record
    b. Authenticate machine-to-machine communication
    c. Ensure all access is logged and monitored
    d. All of the above

25. Which factors are recommended to consider the potential impact of a threat?
    a. Probability, Intent, Capability
    b. Size, Activity, Location, Unit Type, Tactics, Equipment
    c. Size, Activity, Movement, Doctrine, Operations, Command
    d. Confidentiality, Integrity, Availability

26. In the context of DevSecOps, how do you put in place 'just enough' security?
    a. Invest as much as possible to protect assets
    b. Strike a balance between real and perceived exposure
    c. Let the business decide based on the value of its data
    d. Implement countermeasures for all threats

27. Which practice increases an organization's risk profile relative to IAM?
    a. Enabling MFA
    b. Storing secrets outside of vault
    c. Identifying high-risk users
    d. Regularly auditing policies

28. Which of the following practices support DevSecOps?
    a. Implement security as code
    b. Leverage automation
    c. Involve audit and compliance early
    d. All of the above

29. In the context of Westrum's research, which is NOT a characteristic of a generative (performance oriented) culture?
    a. Failure is viewed as a learning opportunity
    b. Cooperation is difficult
    c. New ideas are welcomed
    d. Risks and responsibilities are shared

30. Mr. Jones works for a large organization with extensive compliance requirements but only a limited security budget. He has already hired two senior security experts but must provide routine coverage and integration for 20-30 development teams as well as global operations. What may be a best practice to extend his security coverage at scale?
    a. Divest corporate assets into smaller venture capital considerations
    b. Strict approval processes
    c. Security champions
    d. Policy as Code

31. Which can be used to measure how long a vulnerability or software bug exists before it is identified?
    a. Mean Time to Change (MTTC)
    b. Mean Time to Detect (MTTD)
    c. Mean Time to Recovery (MTTR)
    d. Deployment Frequency

32. Which statement about Emergency Response is INCORRECT?
    a. Carefully selects key stakeholders for process inclusion
    b. Prefers agile responses over documented plans
    c. Functions based on agreed RACI Matrix
    d. Maintains high availability for critical assets

33. Application security's tendency to hand one-off reports to Dev teams outside their normal operating cycles is a bottleneck in an organization's software delivery life cycle. In the context of application security testing, which practice can BEST be used to remove this constraint?
    a. Automatically log findings into issue management
    b. Automate the transfer of data between GRC and issue management
    c. Have Application Support handle high-priority issues
    d. Provide developers real-time findings reports

34. What is the first step to understanding the protection metrics associated with DevSecOps?
    a. Decompose the application
    b. Find the organization's crown jewels
    c. Conduct a source code review
    d. Develop telemetry for all processes

35. Friction can arise when auditors don't understand an organization's new DevOps practices and are unable to use their traditional controls. Which practice would most likely NOT alleviate auditors' objections and concerns?
    a. Integrate auditors into the advice process
    b. Direct auditors to the issue management tool
    c. Build dashboards for the auditors
    d. Provide real-time reporting

36. Which statement about cloud forensics and incident response is INCORRECT?
    a. Emphasis is on live response
    b. Responsibility of the cloud provider
    c. Requires incident response planning
    d. Data capture and workflow can be automated

37. After a series of successful pilots, an organization wants to scale its DevSecOps practices across the enterprise. Which practice should they AVOID?
    a. Use pre-blessed security libraries
    b. Allocate team time to sit and learn together
    c. Automate security testing to promote fast feedback
    d. Create and dictate a clear set of security policies

38. Which statement about separation of duties and DevOps is INCORRECT?
    a. Auditors must redefine controls
    b. DevOps testing helps discover fraud and errors
    c. Developers can submit code to testing pipelines rather than production
    d. DevOps supports the principle of shared responsibilities

39. Which statement about DevSecOps and business transformation is CORRECT?
    a. Security enables transformation by minimizing constraints
    b. Security plays no role in transforming the business
    c. Security and DevOps practices help change how the business functions
    d. Transformation occurs when people make better security decisions

40. The advantage of obtaining a professional certification to validate your learning practice is:
    a. Recognized at multiple levels across the profession
    b. Long lead time
    c. Free drinks at DevOps Days events
    d. Personnel experience including Git projects

Question	Answer	Topic
1	B	DSOF-1 REALIZING DEVSECOPS OUTCOMES
2	D	DSOF-4 INTEGRATING DEVSECOPS STAKEHOLDERS
3	D	DSOF-5 ESTABLISHING DEVSECOPS PRACTICES
4	C	DSOF-6 BEST PRACTICES TO GET STARTED
5	D	DSOF-7 DEVOPS PIPELINES AND CONTINUOUS COMPLIANCE
6	B	DSOF-3 BUILDING A RESPONSIVE DEVSECOPS MODEL
7	A	DSOF-1 REALIZING DEVSECOPS OUTCOMES
8	A	DSOF-4 INTEGRATING DEVSECOPS STAKEHOLDERS
9	B	DSOF-5 ESTABLISHING DEVSECOPS PRACTICES
10	D	DSOF-2 DEFINING THE CYBERTHREAT LANDSCAPE
11	A	DSOF-5 ESTABLISHING DEVSECOPS PRACTICES
12	A	DSOF-6 BEST PRACTICES TO GET STARTED
13	D	DSOF-1 REALIZING DEVSECOPS OUTCOMES
14	A	DSOF-7 DEVOPS PIPELINES AND CONTINUOUS COMPLIANCE
15	B	DSOF-7 DEVOPS PIPELINES AND CONTINUOUS COMPLIANCE
16	D	DSOF-6 BEST PRACTICES TO GET STARTED
17	B	DSOF-6 BEST PRACTICES TO GET STARTED
18	A	DSOF-1 REALIZING DEVSECOPS OUTCOMES
19	D	DSOF-3 BUILDING A RESPONSIVE DEVSECOPS MODEL
20	C	DSOF-5 ESTABLISHING DEVSECOPS PRACTICES
21	C	DSOF-7 DEVOPS PIPELINES AND CONTINUOUS COMPLIANCE
22	C	DSOF-6 BEST PRACTICES TO GET STARTED
23	D	DSOF-3 BUILDING A RESPONSIVE DEVSECOPS MODEL
24	D	DSOF-4 INTEGRATING DEVSECOPS STAKEHOLDERS
25	A	DSOF-2 DEFINING THE CYBERTHREAT LANDSCAPE
26	B	DSOF-5 ESTABLISHING DEVSECOPS PRACTICES
27	B	DSOF-6 BEST PRACTICES TO GET STARTED
28	D	DSOF-3 BUILDING A RESPONSIVE DEVSECOPS MODEL
29	B	DSOF-4 INTEGRATING DEVSECOPS STAKEHOLDERS
30	C	DSOF-5 ESTABLISHING DEVSECOPS PRACTICES
31	B	DSOF-2 DEFINING THE CYBERTHREAT LANDSCAPE
32	B	DSOF-6 BEST PRACTICES TO GET STARTED
33	A	DSOF-3 BUILDING A RESPONSIVE DEVSECOPS MODEL
34	B	DSOF-2 DEFINING THE CYBERTHREAT LANDSCAPE
35	B	DSOF-4 INTEGRATING DEVSECOPS STAKEHOLDERS
36	B	DSOF-5 ESTABLISHING DEVSECOPS PRACTICES
37	D	DSOF-1 REALIZING DEVSECOPS OUTCOMES
38	A	DSOF-5 ESTABLISHING DEVSECOPS PRACTICES
39	C	DSOF-4 INTEGRATING DEVSECOPS STAKEHOLDERS
40	A	DSOF-8 LEARNING USING OUTCOMES

DevSecOps Foundation^{SM} Course:
Value Added Resources

This document provides links to articles and videos related to the DevSecOps Foundation course from the DevOps Institute. This information is provided to enhance your understanding of DevSecOps-related concepts and terms and is not examinable. Of course, there is a wealth of other videos, blogs and case studies on the web.
We welcome suggestions for additions.

Videos Featured in the Course

Module	Title & Description	Link
1: Realizing DevSecOps Outcomes	'DevSecOps: What is It? Why is It Taking Over Security?' with Shannon Lietz (19:18)	https://youtu.be/V9IuDB8ICJM
2: Defining the Cyber Threat Landscape	'The Industrial Cyberthreat Landscape: 2019 Year in Review' with Robert Lee (09:16)	https://youtu.be/3yQrzUEJAkI
3: Building a Responsive DevSecOps Model	'What is DevSecOps' with Dave Farley (19:11)	https://youtu.be/NdvMUcWNlFw
4: Integrating DevSecOps Stakeholders	'Lean and Agile Adoption with the Laloux Culture Model' with Peter Green (09:21)	https://youtu.be/g0Jc5aAJu9g
5: Establishing DevSecOps Practices	'The Rise of DevSecOps' by Yvonne Wassenaar (14:58)	https://youtu.be/LOii0t2fdll
6: Best Practices to Get Started	'Building Security into an Agile Cloud Transformation Project' by Chris Rutter (24:57)	https://youtu.be/clhiT8Le6pk
6: Best Practices to Get Started	'DevSecOps Best Practices with JFrog Platform' with Kat Cosgrove (38:54)	https://youtu.be/wZb2l4ZB4kg
7: DevOps Pipelines and Continuous Compliance	'Overview of DevSecOps' by Nicolas Chaillan (06:24)	https://youtu.be/YpGLUW15JlI
8: Learning Using Outcomes	'Failure and the Third Way' by Aaron Blythe (05:26)	https://www.youtube.com/watch?v=1vn2g_rm-d8

DevSecOps Foundation^{SM} Course: Value Added Resources

DevOps Reports

Report & Link	Writers/Publishers
2020 DevSecOps Community Survey	Sonatype
2020 Global Developer Report	Gitlab
The Accelerate State of DevOps Report 2019	Dr. Nicole Forsgren, Gene Kim & Jez Humble in collaboration with Google Cloud Platform (GCP)
The State of Agile	digital.ai
The State of DevOps Report 2019	Puppet Labs, CircleCI and Splunk
Upskilling: Enterprise DevOps Skills Report 2020	DevOps Institute

DevOps Articles

Relevant Module	Article & Link
1: Realizing DevSecOps Outcomes	The DevSecOps Manifesto. An agile transformation approach to… | by Larry Maccherone | Continuous Agile
1: Realizing DevSecOps Outcomes	CALMS for DevSecOps: Part 1—Why Culture Is Critical
1: Realizing DevSecOps Outcomes	DevSecOps: Why You Need Automation, Fast
1: Realizing DevSecOps Outcomes	DevSecOps—How Lean Improves Performance
1: Realizing DevSecOps Outcomes	Save Time and Avoid Breaches [DevSecOps]
1: Realizing DevSecOps Outcomes	"It's Going to Get Messy": DevSecOps and the Power of Dojos and ChatOps
1: Realizing DevSecOps Outcomes	The 3 Ways of DevSecOps (Part 1) The 3 Ways of DevSecOps (Part 1)

1: Realizing DevSecOps Outcomes	The 3 Ways of DevSECOps by OWASP
1: Realizing DevSecOps Outcomes	How ITIL4 and SRE align with DevOps
1: Realizing DevSecOps Outcomes	Security Should Stop Being a Drag
1: Realizing DevSecOps Outcomes	Security Rituals for Agile Teams
1: Realizing DevSecOps Outcomes	Free Book: Service Management in a DevOps World
1: Realizing DevSecOps Outcomes	DevSecOps and ITIL4 • DevOps Institute
1: Realizing DevSecOps Outcomes	ITIL 4 Managing Professional | ITIL Certifications
1: Realizing DevSecOps Outcomes	Free Book: Service Management in a DevOps World
1: Realizing DevSecOps Outcomes	Beyond The Phoenix Project: Modules 4, 5, 6 –Lean, Safety Culture and Learning Organizations
1: Realizing DevSecOps Outcomes	The Pursuit of Success & Averting Drift into Failure | Sidney Dekker
1: Realizing DevSecOps Outcomes	Safety culture
1: Realizing DevSecOps Outcomes	Create a Culture of Strength: Resilience Engineering
2: Defining the Cyber Threat Landscape	Semaphore: The World's First Telegraph
2: Defining the Cyber Threat Landscape	Rewind - The crooked timber of humanity | 1843

2: Defining the Cyber Threat Landscape	The Confessions of Marcus Hutchins, the Hacker Who Saved the Internet
2: Defining the Cyber Threat Landscape	2020 Roundup Of Cybersecurity Forecasts And Market Estimates
2: Defining the Cyber Threat Landscape	Verizon 2020 Data Breach Investigations Report
2: Defining the Cyber Threat Landscape	2019 Global Cyber Risk Perception Survey
2: Defining the Cyber Threat Landscape	Cyber Coverage Confusion – Risk Management
2: Defining the Cyber Threat Landscape	What Mondelez v. Zurich May Reveal About Cyber Insurance in the Age of Digital Conflict
2: Defining the Cyber Threat Landscape	Full Case Electronic Docket Search
2: Defining the Cyber Threat Landscape	Introduction to the OCTAVE Approach
2: Defining the Cyber Threat Landscape	Threat Modeling: 12 Available Methods
2: Defining the Cyber Threat Landscape	Pushing Left, Like a Boss -Part 6: Threat Modelling
2: Defining the Cyber Threat Landscape	MITRE ATT&CK®
2: Defining the Cyber Threat Landscape	adamshostack/eop: The Elevation of Privilege Threat Modeling Game
2: Defining the Cyber Threat Landscape	OWASP Cornucopia
2: Defining the Cyber Threat Landscape	ENISA Threat Landscape Report 2018 — ENISA

2: Defining the Cyber Threat Landscape	OWASP Top Ten Web Application Security Risks | OWASP
2: Defining the Cyber Threat Landscape	CVE - Common Vulnerabilities and Exposures (CVE)
2: Defining the Cyber Threat Landscape	The Treacherous 12 - Cloud Computing Top Threats in 2016
2: Defining the Cyber Threat Landscape	Book Page
2: Defining the Cyber Threat Landscape	Supply Chain Risk Management Practices for Federal Information Systems and Organizations
2: Defining the Cyber Threat Landscape	Are You Part of a Supply Chain Attack? | Avast
2: Defining the Cyber Threat Landscape	https://thehackernews.com/2018/04/ccleaner-malware-attack.html#email-outer
2: Defining the Cyber Threat Landscape	Software Supply Chain Attacks
2: Defining the Cyber Threat Landscape	Supply Chain Risk Management Practices for Federal Information Systems and Organizations
2: Defining the Cyber Threat Landscape	The CIA triad: Definition, components and examples
2: Defining the Cyber Threat Landscape	Distributed, Immutable, Ephemeral - Diagram
2: Defining the Cyber Threat Landscape	Seven Winning DevSecOps Metrics Security Should Track
2: Defining the Cyber Threat Landscape	Cybermaturity Platform
2: Defining the Cyber Threat Landscape	What is GDPR? The summary guide to GDPR compliance in the UK

2: Defining the Cyber Threat Landscape	Sarbanes-Oxley Act - Summary of Key Provisions
2: Defining the Cyber Threat Landscape	PCI Quick Reference Guide
2: Defining the Cyber Threat Landscape	Cybersecurity
2: Defining the Cyber Threat Landscape	ISO/IEC 27001 — Information security management
2: Defining the Cyber Threat Landscape	Benchmark Corp - Governance Engineering Whitepaper
2: Defining the Cyber Threat Landscape	Continuous Compliance
3: Building a Responsive DevSecOps Model	How to start building a DevSecOps model
3: Building a Responsive DevSecOps Model	DevSecOps
3: Building a Responsive DevSecOps Model	A DevSecOps Guide
3: Building a Responsive DevSecOps Model	DevSecOps Guide
3: Building a Responsive DevSecOps Model	eDiscovery in digital forensic investigations
3: Building a Responsive DevSecOps Model	3 Steps to Ensure Compliance and Audit Success With DevOps
3: Building a Responsive DevSecOps Model	2019 DevSecOps Reference Architectures
3: Building a Responsive DevSecOps Model	OWASP Appsec Pipeline

3: Building a Responsive DevSecOps Model	OWASP DevSecOps Guideline
3: Building a Responsive DevSecOps Model	OWASP Devsecops Maturity Model
4: Integrating DevSecOps Stakeholders	DevSecOps is the Krav Maga of Security — devsecops
4: Integrating DevSecOps Stakeholders	5 Lessons We've Learned Using AWS
4: Integrating DevSecOps Stakeholders	Organizational culture is like an iceberg
4: Integrating DevSecOps Stakeholders	The DevSecOps Manifesto. An agile transformation approach to… | by Larry Maccherone | Continuous Agile
4: Integrating DevSecOps Stakeholders	Decision Making - Home
4: Integrating DevSecOps Stakeholders	What Is Your DevSecOps Manifesto? (Interview with Larry Maccherone)
4: Integrating DevSecOps Stakeholders	The Trust Equation: A Primer
4: Integrating DevSecOps Stakeholders	Erik Erikson's Stages of Psychosocial Development
4: Integrating DevSecOps Stakeholders	DevOps culture: Westrum organizational culture
4: Integrating DevSecOps Stakeholders	Frederic Laloux „Reinventing organizations“
4: Integrating DevSecOps Stakeholders	Rebellious Practices: Make Better Decisions with the Advice Process
4: Integrating DevSecOps Stakeholders	Joy at Work

4: Integrating DevSecOps Stakeholders	CISOs Should Not Report to CIOs
4: Integrating DevSecOps Stakeholders	Dear-Auditor
4: Integrating DevSecOps Stakeholders	Biz-PMO-Dev-QA-Sec-Build-Stage-Ops-Biz
4: Integrating DevSecOps Stakeholders	DoD Enterprise DevSecOps Reference Design
5: Establishing DevSecOps Practices	Bake Security into your SDLC from the beginning
5: Establishing DevSecOps Practices	The leaky pipe of secure coding
5: Establishing DevSecOps Practices	Developer-Centred Security
5: Establishing DevSecOps Practices	Engineering Continuous Security by Marc Hornbeek
5: Establishing DevSecOps Practices	DevOps and Separation of Duties
5: Establishing DevSecOps Practices	Separation of Duties: How to Conform in a DevOps World - XebiaLabs
5: Establishing DevSecOps Practices	Does DevSecOps eliminate the segregation of duties between security and DevOps?
5: Establishing DevSecOps Practices	DevSecOps: Security at the Speed of DevOps
5: Establishing DevSecOps Practices	The Biggest Security Risks Lurking in Your CI/CD Pipeline
5: Establishing DevSecOps Practices	DevSecOps - Secure CI/CD

5: Establishing DevSecOps Practices	Understanding the CI/CD Pipeline: What It Is, Why It Matters
5: Establishing DevSecOps Practices	Cloud Controls Matrix
5: Establishing DevSecOps Practices	Computer Security Resource Center
5: Establishing DevSecOps Practices	The 20 Most Common CASB Use Cases by netskope
5: Establishing DevSecOps Practices	SANS Institute
5: Establishing DevSecOps Practices	Digital defenders: from security geek to C-suite superhero
5: Establishing DevSecOps Practices	Cybersecurity Red Team Versus Blue Team — Main Differences Explained
5: Establishing DevSecOps Practices	Introducing the InfoSec colour wheel — blending developers with red and blue security teams.
5: Establishing DevSecOps Practices	SCYTHE Library: The Purple Team - Organization or Exercise
5: Establishing DevSecOps Practices	OODA loop
6: Best Practices to Get Started	Integrate on-premises AD with Azure - Azure Architecture Center
6: Best Practices to Get Started	Federating multiple Azure AD with single AD FS - Azure
6: Best Practices to Get Started	Integrate on-premises AD domains with Azure AD - Azure Reference Architectures
6: Best Practices to Get Started	What is SAML and How Does it Work?

6: Best Practices to Get Started	Why Organizations Need Adaptive Multi-factor Authentication (MFA)
6: Best Practices to Get Started	The secret to DevOps secrets management
6: Best Practices to Get Started	Managing Secrets in DevOps: A Maturity Model
6: Best Practices to Get Started	DevOps & Proliferation of Secrets [Keeping Data Safe]
6: Best Practices to Get Started	What is encryption and how does it protect your data?
6: Best Practices to Get Started	4 tips for a successful cyber threat intelligence program
6: Best Practices to Get Started	Developing a Cyber Threat Intelligence Program
6: Best Practices to Get Started	The Impact of Software Security Practice Adoption Quantified
7: DevOps Pipelines and Continuous Compliance	Continuous Integration
7: DevOps Pipelines and Continuous Compliance	ContinuousDelivery
7: DevOps Pipelines and Continuous Compliance	Avast fights off cyber-espionage attempt, Abiss | Avast
7: DevOps Pipelines and Continuous Compliance	Container Security | Prisma
7: DevOps Pipelines and Continuous Compliance	NCR Attains Security & PCI Compliance For Container-Based Applications
7: DevOps Pipelines and Continuous Compliance	Security information and event management

7: DevOps Pipelines and Continuous Compliance	2019 DevSecOps Reference Architectures
8: Learning Using Outcomes	The Three Ways: The Principles Underpinning DevOps
8: Learning Using Outcomes	DevSecOps Days
8: Learning Using Outcomes	SKILup Days by DevOps Institute • DevOps Institute
8: Learning Using Outcomes	The Evolution of DevSecOps [Interview with Marc Cluet]
8: Learning Using Outcomes	Codeup | The Premier Career Accelerator in Texas
8: Learning Using Outcomes	https://careerkarma.com/rankings/best-cyber-security-bootcamps
8: Learning Using Outcomes	How to Lead More Effectively with the SCARF Model
8: Learning Using Outcomes	THE 50 BEST UNIVERSITIES FOR CYBER SECURITY AND INFORMATION ASSURANCE
8: Learning Using Outcomes	10 Tips for a Successful Lunch & Learn
8: Learning Using Outcomes	To Receive, You Must Give: Creating Value as a Mentee
8: Learning Using Outcomes	A Rising Tide Lifts All Boats: DevOps Dojo Stories from DOES London 2020
8: Learning Using Outcomes	DevOps Dojo: What On Earth Is That?
8: Learning Using Outcomes	Dojo Consortium | About the Dojo Consortium

8: Learning Using Outcomes	Getting Started with Dojos
8: Learning Using Outcomes	The Target Dojo
8: Learning Using Outcomes	Security Chaos Engineering: A new paradigm for cybersecurity
8: Learning Using Outcomes	Netflix/security_monkey: Security Monkey monitors AWS, GCP, OpenStack, and GitHub orgs for assets and their changes over time.
8: Learning Using Outcomes	GitHub - linki/chaoskube
8: Learning Using Outcomes	Chaos Monkey Alternatives for Kubernetes
8: Learning Using Outcomes	Building Predictive Security into DevSecOps [Interview with Aaron Rinehart, Part 2]
8: Learning Using Outcomes	Full-Stack Agile - The Sprint Review (Scrum)
8: Learning Using Outcomes	Cyber Warrior Network – Cyber Warfare AI company that specializes in securely connecting Cyber Warriors, Employers, Educators & Trainers to build Skilled Cyber Talent Pipelines ready to combat cybersecurity threats.
8: Learning Using Outcomes	Secure coding solutions for Developers
8: Learning Using Outcomes	Games for Developer-Centred Security
8: Learning Using Outcomes	2019 IT Security Employment Outlook: The Hottest Skills and Markets
8: Learning Using Outcomes	Agile Transformation at Ericsson
8: Learning Using Outcomes	DevOps: fueling the evolution toward 5G networks

8: Learning Using Outcomes	Become a DevOps Institute Community Member
8: Learning Using Outcomes	SKILup Days by DevOps Institute • DevOps Institute
8: Learning Using Outcomes	devsecops | GitHub
8: Learning Using Outcomes	DevSecOps and the 4th Industrial Revolution
8: Learning Using Outcomes	2020 Upskilling: Enterprise DevOps Skills Report • DevOps Institute

WebSites

Title	Link
Agile Alliance	https://www.agilealliance.org/glossary
Agile Manifesto	https://agilemanifesto.org/
Beyond Budgeting	https://bbrt.org/
DevOps Games	https://devopsgames.com/
DevOps Institute	https://devopsinstitute.com/
DevOps Topologies	https://web.devopstopologies.com/
DevOps.com	https://devops.com/
DevOpsDays	https://www.devopsdays.org/
DevSecOps.org	https://www.devsecops.org/
DevSecOps Reference Architectures (Sonatype)	https://www.sonatype.com/devsecops-reference-architectures
IT Revolution	https://itrevolution.com/
Lean Change Canvases	https://leanchange.org/resources/canvases/

LeanStack	https://leanstack.com/leancanvas
Large Scale Scrum (LeSS)	https://less.works/
SAFe	https://www.scaledagileframework.com
Scaled Agile Framework	https://www.scaledagileframework.com/devops/
Scrum.org	https://www.scrum.org/
Scrum Guide	https://www.scrumguides.org/scrum-guide.html
Sidney Dekker	https://sidneydekker.com/
State of Agile	https://stateofagile.com/

DevOps & Engineering Blogs

Blog	Link
Ahmad Iqbal Ali	Git Strategies for DevOps
AirBNB Engineering & Data Science	https://medium.com/airbnb-engineering
Atlassian	Git cheat sheet
Backstage Blog (SoundCloud)	https://developers.soundcloud.com/blog/
BlackRock Blog	http://rockthecode.io/
code.flickr.com	http://code.flickr.net/
Daniel Miessler	https://danielmiessler.com/blog/difference-cve-cwe
Carnegie Mellon University	https://insights.sei.cmu.edu/blog/10-types-of-application-security-testing-tools-when-and-how-to-use-them/
DEFRA Digital	https://defradigital.blog.gov.uk/
Deliveroo Engineering Team	https://deliveroo.engineering/
DevOps Institute	https://www.devopsinstitute.com/podcast-ep49-joel-kruger-struggling-musician-turned-devops-professional/
DevOps Institute	https://devops.com/how-to-become-a-devsecops-engineer/

Dropbox Tech Blog	https://blogs.dropbox.com/tech/
eBay Tech Blog	https://www.ebayinc.com/stories/blogs/tech/
Etsy - Code as Craft	https://codeascraft.com/
Eventbrite Engineering	https://www.eventbrite.com/engineering/
Facebook Engineering	https://www.facebook.com/Engineering
Github	Git Handbook
Github Engineering	https://githubengineering.com/
Google Developers	https://developers.googleblog.com/
Heroku Engineering	https://blog.heroku.com/engineering
HubSpot Engineering	https://product.hubspot.com/blog/topic/engineering
Instagram Engineering	http://instagram-engineering.tumblr.com/
Intra Links	https://www.intralinks.com/blog/2016/01/information-rights-management-irm-need
ISO	https://www.iso.org/iso-31000-risk-management.html
Kickstarter Engineering	https://kickstarter.engineering/
LinkedIn Engineering	https://engineering.linkedin.com/blog
Monzo Technology	https://monzo.com/blog/technology/
Moonpig Engineering	https://engineering.moonpig.com/
Netflix TechBlog	https://medium.com/netflix-techblog
PayPal Engineering	https://www.paypal-engineering.com/
Pinterest Engineering	https://medium.com/@Pinterest_Engineering
Revolut Engineering	https://blog.revolut.com/tag/engineering/
Ruggard Software	https://ruggedsoftware.org/
Salesforce Engineering	https://engineering.salesforce.com/
Slack Engineering	https://slack.engineering/

Target Tech	http://target.github.io/devops/the-dojo
Ticketmaster Technology	https://tech.ticketmaster.com/category/devops/
TechTarget	https://whatis.techtarget.com/definition/data-loss-prevention-DLP
Trainline Engineering	https://engineering.thetrainline.com/
Uber Engineering	https://eng.uber.com/
VENAFI	https://www.venafi.com/blog/why-we-need-devsecops-interview-shannon-lietz
Vimeo Engineering	https://medium.com/vimeo-engineering-blog
Zapier Engineering	https://zapier.com/engineering/

Additional Videos of Interest

Relevant Module	Title	Link
1: Realizing DevSecOps Outcomes	'The (Short) History of DevOps' by Damon Edwards (11:47)	https://youtu.be/o7-IuYS0iSE
1: Realizing DevSecOps Outcomes	'10+ Deploys Per Day: Dev and Ops Cooperation at Flickr' by John Allspaw and Paul Hammond (46:21)	https://youtu.be/LdOe18KhtT4
1: Realizing DevSecOps Outcomes	'What is Scrum? | Agile' by Navin Reddy (9:46)	https://youtu.be/oTZd2vo3FQU
1: Realizing DevSecOps Outcomes	'Convergence Of Safety Culture And Lean: Lessons From The Leaders' by Sidney Dekker, Steven Spear, and Richard Cook (31:06)	https://youtu.be/CFMJ3V4VakA
7: DevOps Pipelines and Continuous Compliance	'Continuous Compliance and DevSecOps in Times of GDPR, HIPAA and SOX' by Torsten Volk and Anders Wallgren (1:00:48)	https://www.youtube.com/watch?v=ZSj46aIbkOI

DevOps Books

Title	Author	Link
Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations	Nicole Forsgren PHD, Jez Humble & Gene Kim	https://itrevolution.com/book/accelerate/
Agile Application Security	Laura Bell, Michael Brunton-Spall, Rich Smith, Jim Bird	https://www.amazon.com/Agile-Application-Security-Enabling-Continuous/dp/1491938846
Beyond The Phoenix Project	Gene Kim and Jez Humble	https://itrevolution.com/book/beyond-phoenix-project/
Building a Modern Security Program	Zane Lackey, Rebecca Huehls	https://www.oreilly.com/library/view/building-a-modern/9781492044680/
Continuous Delivery	Jez Humble and Dave Farley	https://www.amazon.com/dp/0321601912?tag=contindelive-20
Countdown to Zero Day	Kim Zetter	https://www.amazon.com/Countdown-Zero-Day-Stuxnet-Digital/dp/0770436196/ref=sr_1_1?dchild=1&keywords=countdown+to+zero+day&qid=1594125330&sr=8-1
Cyber War Will Not Take Place (Conflict Classics)	Thomas Rid	https://www.amazon.com/Cyber-Will-Place-Conflict-Classics/dp/0190660716/ref=sr_1_1?dchild=1&keywords=cyberwar+will+not+take+place&qid=1594125574&sr=8-1
Dark Territory: The Secret History of Cyber War	Fred M. Kaplan	https://www.amazon.com/Dark-Territory-Secret-History-Cyber/dp/1476763267/ref=sr_1

		_1?dchild=1&keywords=dark+territory&qid=1594125407&sr=8-1
DevOps for the Modern Enterprise	Mirco Hering	https://itrevolution.com/book/devops_modern_enterprise/
DevOpsSec	Jim Bird	https://www.oreilly.com/library/view/devopssec/9781491971413/
Engineering DevOps	Marc Hornbeek	https://www.amazon.ca/dp/1543989616?slotNum=52&linkCode=g12&imprToken=bTFS.C1CuXwJttJhylz-VQ&creativeASIN=1543989616&tag=uuid0a1-20
Hands-On Security in DevOps	Tony Hsu	https://www.amazon.com/Hands-Security-DevOps-continuous-deployment/dp/1788995503
Lean IT	Steven C Bell and Michael A Orzen	https://www.amazon.com/Lean-Enabling-Sustaining-Your-Transformation/dp/1439817561
From Project to Product	Dr. Mik Kersten	https://itrevolution.com/book/project-to-product/
Site Reliability Engineering	Niall Richard Murphy, Betsy Beyer and Chris Jones	https://www.amazon.com/Site-Reliability-Engineering-Production-Systems/dp/149192912X
Securing DevOps	Julien Vehent	https://www.manning.com/books/securing-devops?a_aid=securingdevops&a_bid=1353bcd8
Tallinn Manual 2.0 on the International Law Applicable to Cyber Operations	Michael N. Schmitt	https://www.amazon.com/Tallinn-Manual-International-Applicable-Operations/dp/1316630374/ref=sr_1_1?dchild=1&keywords=t

		allinn+manual&qid=1594125245&sr=8-1
The Art of Business Value	Mark Schwartz	https://itrevolution.com/book/the-art-of-business-value/
The DevOps Handbook	Gene Kim, Jez Humble, Patrick Debois & John Willis	https://itrevolution.com/book/the-devops-handbook/
The Phoenix Project	Kevin Behr, George Spafford and Gene Kim	https://itrevolution.com/book/the-phoenix-project/
The Unicorn Project	Gene Kim	https://itrevolution.com/book/the-unicorn-project/

Case Stories Featured in the Course

Company	Module	Link
Aetna	1: Realizing DevSecOps Outcomes	• https://techbeacon.com/security/how-one-healthcare-giant-stays-focused-application-security • https://youtu.be/8iSvTmpe4e8
Maersk	2: Defining the Cyber Threat Landscape	• https://deloitte.wsj.com/cio/2019/07/23/devsecops-a-steppingstone-to-maersks-future/ • https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/
NCR	3: Building a Responsive DevSecOps Model	• https://www.aquasec.com/customers/ncr-attains-security-pci-compliance-for-its-container-based-applications/ • https://youtu.be/I1GHeebvqPw • https://youtu.be/QvHUYhLebDc • https://techbeacon.com/security/application-security-your-career-5-key-areas-focus • https://techbeacon.com/security/5-ways-scale-your-app-sec-program

US Department of Defense	4: Integrating DevSecOps Stakeholders	• https://dodcio.defense.gov/Portals/0/Documents/DoD%20Enterprise%20DevSecOps%20Reference%20Design%20v1.0_Public%20Release.pdf?ver=2019-09-26-115824-583 • https://www.fedscoop.com/welcome/?dod-looks-scale-devsecops-container-use-across-department&id=33853 • https://www.afcea.org/content/defense-departments-devsecops-initiative-move • https://www.venafi.com/blog/us-dod-reference-design-devsecops-interview-nicolas-chaillan • https://www.infoq.com/news/2020/06/defense-department-devsecops/ • https://youtu.be/rJN-CtPbpjY
Comcast	5: Establishing DevSecOps Practices	• http://www.telcotransformation.com/author.asp?section_id=422&doc_id=723953 • https://www.venafi.com/blog/what-your-devsecops-manifesto-interview-larry-maccherone • https://devops.com/downloads/comcast-business-strategy-hinges-on-devops/ • https://medium.com/continuous-agile/the-devsecops-manifesto-94579e0eb716 • https://www.infoq.com/presentations/devsecops/ • https://www.rsaconference.com/experts/larry-maccherone#:~:text=Larry%20Maccherone%20is%20an%20industry,on%20DevSecOps%2C%20Agile%20and%20Analytics.&text=Maccherone%20has%20also%20served%20as,Los%20Alamos%20National%20Labs%20Fellow. • https://devops.com/downloads/comcast-business-strategy-hinges-on-devops/
Sentara Healthcare	6: Best Practices to Get Started	• https://www.bankinfosecurity.com/interviews/dan-bowden-i-4369
Dropbox	7: DevOps Pipelines and Continuous Compliance	• https://dropbox.tech/security/how-dropbox-security-builds-better-tools-for-threat-detection-and-incident-response
Ericsson	8: Learning Using Outcomes	• https://www.infoq.com/articles/agile-transformation-ericsson/ • https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/devops-fueling-the-evolution-toward-5g-networks

DevOps Institute

DEVOPS GLOSSARY OF TERMS

This glossary is provided for reference only as it contains key terms that may or may not be examinable.

DevOps Glossary of Terms

Term	Definition
12-Factor App Design	A methodology for building modern, scalable, maintainable software-as-a-service applications.
2-Factor *or* 2-Step Authentication	Two-Factor Authentication, also known as 2FA or TFA or Two-Step Authentication is when a user provides two authentication factors; usually, firstly a password and then a second layer of verification such as a code texted to their device, shared secret, physical token, or biometrics.
A/B Testing	Deploy different versions of an EUT to different customers and let the customer feedback determine which is best.
A3 Problem Solving	A structured problem-solving approach that uses a lean tool called the A3 Problem-Solving Report. The term "A3" represents the paper size historically used for the report (a size roughly equivalent to 11" x 17").
Access Management	Granting an authenticated identity access to an authorized resource (e.g., data, service, environment) based on defined criteria (e.g., a mapped role), while preventing unauthorized identity access to a resource.
Access Provisioning	Access provisioning is the process of coordinating the creation of user accounts, e-mail authorizations in the form of rules and roles, and other tasks such as provisioning of physical resources associated with enabling new users to systems or environments.
Administration Testing	The purpose of the test is to determine if an End User Test (EUT) is able to process administration tasks as expected.
Advice Process	Any person making a decision must seek advice from everyone meaningfully affected by the decision and people with expertise in the matter. Advice received must be taken into consideration, though it does not have to be accepted or followed. The objective of the advice process is not to form a consensus, but to inform the decision-maker so that they can make the best decision possible. Failure to follow the advice process undermines trust and unnecessarily introduces risk to the business.
Agile	A work management method for complex endeavors that divides tasks into small "sprints" of work with frequent reassessment and adaptation of plans.

Agile (adjective)	Able to move quickly and easily; well-coordinated. Able to think and understand quickly; able to solve problems and have new ideas.
Agile Coach	Help teams master Agile development and DevOps practices; enables productive ways of working and collaboration.
Agile Enterprise	A fast-moving, flexible, and robust company capable of rapid response to unexpected challenges, events, and opportunities.
Agile Manifesto	A formal proclamation of values and principles to guide an iterative and people-centric approach to software development. http://agilemanifesto.org
Agile Portfolio Management	Involves evaluating in-flight projects and proposed future initiatives to shape and govern the ongoing investment in projects and discretionary work. CA's Agile Central and VersionOne are examples.
Agile Practice Owner	A role accountable for the overall quality of a service management practice and owner of the Practice Backlog.
Agile Principles	The twelve principles that underpin the Agile Manifesto.
Agile Process	Delivers "just enough" structure and control to enable the organization to achieve its service outcomes in the most expeditious, effective, and efficient way possible. It is easy to understand, easy to follow, and prizes its collaboration and outcomes more than its artifacts.
Agile Process Engineering	An iterative and incremental approach to designing a process with short, iterative designs of potentially shippable process increments or microprocesses.
Agile Process Improvement	Ensures that IT Service Management agility introduced through Agile Process Engineering is continually reviewed and adjusted as part of IT Service Management's commitment to continual improvement.
Agile Service Management	A framework that ensures that ITSM processes reflect Agile values and are designed with "just enough" control and structure in order to effectively and efficiently deliver services that facilitate customer outcomes when and how they are needed.
Agile Service Management Artifacts	Practice Backlog, Sprint Backlog, Increment
Agile Service Management Events	Practice/microprocess Planning, The Sprint, Sprint Planning, Process Standup, Sprint Review, Sprint Retrospective

Agile Service Management Roles	Agile Practice Owner, Agile Service Management Team, Agile Service Manager
Agile Service Management Team	A team of at least 3 people (including a customer or practitioner) that is accountable for a single microprocess or a complete service management practice.
Agile Service Manager	An Agile Service Management subject matter expert who is the coach and protector of the Agile Service Management Team.
Agile Software Development	Group of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. Usually applied using the Scrum or Scaled Agile Framework approach.
Amazon Web Services (AWS)	Amazon Web Services (AWS) is a secure cloud services platform, offering compute power, database storage, content delivery, and other functionality to help businesses scale and grow.
Analytics	Test results processed and presented in an organized manner in accordance with analysis methods and criteria.
Andon	A system gives an assembly line worker the ability, and moreover the empowerment, to stop production when a defect is found, and immediately call for assistance.
Anti-pattern	A commonly reinvented but poor solution to a problem.
Anti-fragility	Antifragility is a property of systems that increases its capability to thrive as a result of stressors, shocks, volatility, noise, mistakes, faults, attacks, or failures.
API Testing	The purpose of the test is to determine if an API for an EUT functions as expected.
Application Performance Management (APM)	APM is the monitoring and management of the performance and availability of software applications. APM strives to detect and diagnose complex application performance problems to maintain an expected level of service.
Application Programming Interface (API)	A set of protocols used to create applications for a specific OS or as an interface between modules or applications.

Application Programming Interface (API) Testing	The purpose of the test is to determine if an API for an EUT functions as expected.
Application Release	Controlled continuous delivery pipeline capabilities including automation (release upon code commit).
Application Release Automation (ARA) or Orchestration (ARO)	Controlled continuous delivery pipeline capabilities including automation (release upon code commit), environment modeling (end-to-end pipeline stages, and deploy application binaries, packages, or other artifacts to target environments), and release coordination (project, calendar, and scheduling management, integrate with change control and/or IT service support management).
Application Test-Driven Development (ATDD)	Acceptance Test-Driven Development (ATDD) is a practice in which the whole team collaboratively discusses acceptance criteria, with examples, and then distills them into a set of concrete acceptance tests before development begins.
Application Testing	The purpose of the test is to determine if an application is performing according to its requirements and expected behaviors.
Application Under Test (AUT)	The EUT is a software application. E.g. Business application is being tested.
Architecture	The fundamental underlying design of computer hardware, software, or both in combination.
Artifact	Any element in a software development project including documentation, test plans, images, data files, and executable modules.
Artifact Repository	Store for binaries, reports, and metadata. Example tools include JFrog Artifactory, Sonatype Nexus.
Attack path	The chain of weaknesses a threat may exploit to achieve the attacker's objective. For example, an attack path may start by compromising a user's credentials, which are then used in a vulnerable system to escalate privileges, which in turn is used to access a protected database of information, which is copied out to an attacker's own server(s).
Audit Management	The use of automated tools to ensure products and services are auditable, including keeping audit logs of build, test and deploy activities, auditing configurations, and users, as well as log files from production operations.

Authentication	The process of verifying an asserted identity. Authentication can be based on what you know (e.g., password or PIN), what you have (token or one-time code), what you are (biometrics), or contextual information.
Authorization	The process of granting roles to users to have access to resources.
Auto-DevOps	Auto DevOps brings DevOps best practices to your project by automatically configuring software development lifecycles. It automatically detects, builds, tests, deploys, and monitors applications.
Auto-scaling	The ability to automatically and elastically scale and de-scale infrastructure depending on traffic and capacity variations while maintaining control of costs.
Automated rollback	If a failure is detected during a deployment, an operator (or an automated process) will verify the failure and roll back the failing release to the previous known working state.
Availability	Availability is the proportion of time a system is in a functioning condition and therefore available (to users) to be used.
Backdoor	A backdoor bypasses the usual authentication used to access a system. Its purpose is to grant the cybercriminals future access to the system even if the organization has remediated the vulnerability initially used to attack the system.
Backlog	Requirements for a system expressed as a prioritized list of product backlog items usually in the form of 'User Stories'. The product backlog is prioritized by the Product Owner and should include functional, non-functional, and technical team-generated requirements.
Basic Security Hygiene	A common set of minimum-security practices that must be applied to all environments without exception. Practices include basic network security (firewalls and monitoring), hardening, vulnerability and patch management, logging and monitoring, basic policies and enforcement (may be implemented under a "policies as code" approach), and identity and access management.
Batch Sizes	Refers to the volume of features involved in a single code release.
Bateson Stakeholder Map	A tool for mapping stakeholder's engagement with the initiative in progress.
Behavior Driven Development (BDD)	Test cases are created by simulating an EUT's externally observable inputs, and outputs. Example tool: Cucumber.

Beyond Budgeting	A management model that looks beyond command-and-control towards a more empowered and adaptive state.
Black-Box	Test case only uses knowledge of externally observable behaviors of an EUT.
Blameless post mortems	A process through which engineers whose actions have contributed to a service incident can give a detailed account of what they did without fear of punishment or retribution.
Blast Radius	Used for impact analysis of service incidents. When a particular IT service fails, the users, customers, other dependent services that are affected.
Blue/Green Testing or Deployments	Taking software from the final stage of testing to live production using two environments labeled Blue and Green. Once the software is working in the green environment, switch the router so that all incoming requests go to the green environment - the blue one is now idle.
Bug	An error or defect in software that results in an unexpected or system-degrading condition.
Bureaucratic Culture	Bureaucratic organizations are likely to use standard channels or procedures which may be insufficient in a crisis (Westrum).
Bursting	Public cloud resources are added as needed to temporarily increase the total computing capacity of a private cloud.
Business Case	Justification for a proposed project or undertaking on the basis of its expected commercial benefit.
Business Continuity	Business continuity is an organization's ability to ensure operations and core business functions are not severely impacted by a disaster or unplanned incident that takes critical services offline.
Business Transformation	Changing how the business functions. Making this a reality means changing culture, processes, and technologies in order to better align everyone around delivering on the organization's mission.
Business Value	In management, an informal term that includes all forms of value that determine the health and well-being of the firm in the long run.
Cadence	Flow or rhythm of events.

CALMS Model	Considered the pillars or values of DevOps: Culture, Automation, Lean, Measurement, Sharing (as put forth by John Willis, Damon Edwards, and Jez Humble).
Canary Testing	A canary (also called a canary test) is a push of code changes to a small number of end-users who have not volunteered to test anything. Similar to incremental rollout, it is where a small portion of the user base is updated to a new version first. This subset, the canaries, then serve as the proverbial "canary in the coal mine". If something goes wrong then a release is rolled back and only a small subset of the users are impacted.
Capacity	An estimate of the total amount of engineering time available for a given Sprint.
Capacity Test	The purpose of the test is to determine if the EUT can handle expected loads such as number of users, number of sessions, aggregate bandwidth.
Capture-Replay	Test cases are created by capturing live interactions with the EUT, in a format that can be replayed by a tool. E.g. Selenium
Carrots	Positive incentives, for encouraging and rewarding desired behaviors.
Chain of Goals	A method designed by Roman Pichler of ensuring that goals are linked and shared at all levels through the product development process.
Change	Addition, modification, or removal of anything that could have an effect on IT services. (ITIL® definition)
Change Failure Rate	A measure of the percentage of failed/rolled back changes.
Change Fatigue	A general sense of apathy or passive resignation towards organizational changes by individuals or teams.
Change Lead Time	A measure of the time from a request for a change to the delivery of the change.
Change Leader Development Model	Jim Canterucci's model for five levels of change leader capability.
Change Management	The process that controls all changes throughout their lifecycle. (ITIL definition)

Change Management (Organizational)	An approach to shifting or transitioning individuals, teams & organizations from a current state to a desired future state. Includes the process, tools & techniques to manage the people-side of change to achieve the required business outcome(s).
Change-based Test Selection Method	Tests are selected according to a criterion that matches attributes of tests to attributes of the code that is changed in a build.
Chaos Engineering	The discipline of experimenting on a software system in production in order to build confidence in the system's capability to withstand turbulent and unexpected conditions.
Chapter Lead	A squad line manager in the Spotify model who is responsible for traditional people management duties is involved in day-to-day work, and grows individual and chapter competence.
Chapters	A small family of people having similar skills and who work within the same general competency area within the same tribe. Chapters meet regularly to discuss challenges and areas of expertise in order to promote sharing, skill development, re-use, and problem-solving.
ChatOps	An approach to managing technical and business operations (coined by GitHub) that involves a combination of group chat and integration with DevOps tools. Example tools include Atlassian HipChat/Stride, Microsoft Teams, Slack.
Check-in	The action of submitting a software change into a system version management system.
CI Regression Test	A subset of regression tests that are run immediately after a software component is built. Same as Smoke Test.
Clear-Box	Same as Glass-Box Testing and White-Box Testing.
Cloud Computing	The practice of using remote servers hosted on the internet to host applications rather than local servers in a private data center.
Cloud-Native	Native cloud applications (NCA) are designed for cloud computing.

Cloudbees	Cloudbees is a commercially supported proprietary automation framework tool that works with and enhances Jenkins by providing enterprise levels support and add-on functionality.
Cluster Cost Optimization	Tools like Kubecost, Replex, Cloudability use monitoring to analyze container clusters and optimize the resource deployment model.
Cluster Monitoring	Tools that let you know the health of your deployment environments running in clusters such as Kubernetes.
Clustering	A group of computers (called nodes or members) work together as a cluster connected through a fast network acting as a single system.
Code Coverage	A measure of white box test coverage by counting code units that are executed by a test. The code unit may be a code statement, a code branch, or control path or data path through a code module.
Code Quality	See also static code analysis, Sonar and Checkmarks are examples of tools that automatically check the seven main dimensions of code quality – comments, architecture, duplication, unit test coverage, complexity, potential defects, language rules.
Code Repository	A repository where developers can commit and collaborate on their code. It also tracks historical versions and potentially identifies conflicting versions of the same code. Also referred to as "repository" or "repo."
Code Review	Software engineers inspect each other's source code to detect coding or code formatting errors.
Cognitive Bias	Cognitive bias is a limitation in objective thinking that is caused by the tendency for the human brain to perceive information through a filter of personal experience and preferences: a systematic pattern of deviation from norm or rationality in judgment.
Collaboration	People jointly working with others towards a common goal.
Collaborative Culture	A culture that applies to everyone which incorporates an expected set of behaviors, language, and accepted ways of working with each other reinforcement by leadership.
Compatibility Test	Test with the purpose to determine if an EUT interoperates with another EUT such as peer-to-peer applications or protocols.

Configuration Management	Configuration management (CM) is a systems engineering process for establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life.
Conformance Test	The purpose of the test is to determine if an EUT complies with a standard.
Constraint	Limitation or restriction; something that constrains. See also *bottleneck*.
Container	A way of packaging software into lightweight, stand-alone, executable packages including everything needed to run it (code, runtime, system tools, system libraries, settings) for development, shipment, and deployment.
Container Network Security	Used to prove that any app that can be run on a container cluster with any other app can be confident that there is no unintended use of the other app or any unintended network traffic between them.
Container Registry	Secure and private registry for Container images. Typically allowing for easy upload and download of images from the build tools. Docker Hub, Artifactory, Nexus are examples.
Container Scanning	When building a Container image for your application, tools can run a security scan to ensure it does not have any known vulnerability in the environment where your code is shipped. Blackduck, Synopsis, Synk, Claire, and Klar are examples.
Continual Service Improvement (CSI)	One of the ITIL Core publications and a stage of the service lifecycle.
Continuous Delivery (CD)	A methodology that focuses on making sure software is always in a releasable state throughout its lifecycle.
Continuous Delivery (CD) Architect	A person who is responsible to guide the implementation and best practices for a continuous delivery pipeline.
Continuous Delivery Pipeline	A continuous delivery pipeline refers to the series of processes that are performed on product changes in stages. A change is injected at the beginning of the pipeline. A change may be new versions of code, data, or images for applications. Each stage processes the artifacts resulting from the prior stage. The last stage results in deployment to production.

Continuous Delivery Pipeline Stage	Each process in a continuous delivery pipeline. These are not standard. Examples are Design: determine implementation changes; Creation: implement an unintegrated version of design changes; Integration: merge
Continuous Deployment	A set of practices that enable every change that passes automated tests to be automatically deployed to production.
Continuous Flow	Smoothly moving people or products from the first step of a process to the last with minimal (or no) buffers between steps.
Continuous Improvement	Based on Deming's Plan-Do-Check-Act, a model for ensuring ongoing efforts to improve products, processes, and services.
Continuous Integration (CI)	A development practice that requires developers to merge their code into trunk or master ideally at least daily and perform tests (i.e. unit, integration, and acceptance) at every code commit.
Continuous Integration Tools	Tools that provide an immediate feedback loop by regularly merging, building, and testing code. Example tools include Atlassian Bamboo, Jenkins, Microsoft VSTS/Azure DevOps, TeamCity.
Continuous Monitoring (CM)	This is a class of terms relevant to logging, notifications, alerts, displays, and analysis of test results information.
Continuous Testing (CT)	This is a class of terms relevant to the testing and verification of an EUT in a DevOps environment.
Conversation Café	Conversation Cafés are open, hosted conversations in cafés as well as conferences and classrooms—anywhere people gather to make sense of our world.
Conway's Law	Organizations that design systems are constrained to produce designs that are copies of the communication structures of these organizations.
Cooperation vs. Competition	The key cultural value shift toward being highly collaborative and cooperative, and away from internal competitiveness and divisiveness.

COTS	Commercial-off-the-shelf solution
Critical Success Factor (CSF)	Something that must happen for an IT service, process, plan, project or other activity to succeed.
Cultural Iceberg	A metaphor that visualizes the difference between observable (above the water) and non-observable (below the waterline) elements of culture.
Culture (Organizational Culture)	The values and behaviors that contribute to the unique psychosocial environment of an organization.
Cumulative Flow Diagram	A cumulative flow diagram is a tool used in agile software development and lean product development. It is an area graph that depicts the quantity of work in a given state, showing arrivals, time in queue, quantity in a queue, and departure.
Current State Map	A form of value stream map that helps you identify how the current process works and where the disconnects are.
Customer Reliability Engineer (CRE)	CRE is what you get when you take the principles and lessons of SRE and apply them to customers.
Cycle Time	A measure of the time from the start of work to ready for delivery.
Daily Scrum	Daily timeboxed event of 15 minutes or less for the Team to replan the next day of work during a Sprint.
Dashboard	Graphical display of summarized data e.g., deployment frequency, velocity, test results.
DAST (Dynamic Application Security Testing)	Dynamic application security testing (DAST) is a process of testing an application or software product in an operating state.
Data Loss Protection (DLP)	Tools that prevent files and content from being removed from within a service environment or organization.
Database Reliability Engineer (DBRE)	A person responsible for keeping database systems that support all user-facing services in production running smoothly.

Defect Density	The number of faults found in a unit E.g. # defects per KLOC, # defects per change.
Definition of Done	A shared understanding of expectations that an Increment or backlog item must live up to.
Delivery Cadence	The frequency of deliveries. E.g. # deliveries per day, per week, etc.
Delivery Package	Set of release items (files, images, etc.) that are packaged for deployment.
Deming Cycle	A four-stage cycle for process management, attributed to W. Edwards Deming. Also called Plan-Do-Check-Act (PDCA).
Dependency Firewall	Many projects depend on packages that may come from unknown or unverified providers, introducing potential security vulnerabilities. There are tools to scan dependencies but that is after they are downloaded. These tools prevent those vulnerabilities from being downloaded to begin with.
Dependency Proxy	For many organizations, it is desirable to have a local proxy for frequently used upstream images/packages. In the case of CI/CD, the proxy is responsible for receiving a request and returning the upstream image from a registry, acting as a pull-through cache.
Dependency Scanning	Used to automatically find security vulnerabilities in your dependencies while you are developing and testing your applications. Synopsys, Gemnasium, Retire.js, and bundler-audit are popular tools in this area.
Deployment	The installation of a specified version of software to a given environment (e.g., promoting a new build into production).
Design for Testability	An EUT is designed with features that enable it to be tested.
Design Principles	Principles for designing, organizing, and managing a DevOps delivery operating model.
Dev	Individuals involved in software development activities such as application and software engineers.

Developer (Dev)	An individual who has the responsibility to develop changes for an EUT. Alternate: Individuals involved in software development activities such as application and software engineers.
Development Test	Ensuring that the developer's test environment is a good representation of the production test environment.
Device Under Test (DUT)	The DUT is a device (e.g. router or switch) being tested.
DevOps	A cultural and professional movement that stresses communication, collaboration, and integration between software developers and IT operations professionals while automating the process of software delivery and infrastructure changes. It aims at establishing a culture and environment where building, testing, and releasing software, can happen rapidly, frequently, and more reliably." (Wikipedia)
DevOps Coach	Help teams master Agile development and DevOps practices; enables productive ways of working and collaboration.
DevOps Infrastructure	The entire set of tools and facilities that make up the DevOps system. Includes CI, CT, CM, and CD tools.
DevOps Kaizen	Kaizen is a Japanese word that closely translates to "change for better," the idea of continuous improvement—large or small—involving all employees and crossing organizational boundaries. Damon Edwards' DevOps Kaizen shows how making small, incremental improvements (little J's) has an improved impact on productivity long term.
DevOps Pipeline	The entire set of interconnected processes that make up a DevOps Infrastructure.
DevOps Score	A metric showing DevOps adoption across an organization and the corresponding impact on delivery velocity.
DevOps Toolchain	The tools needed to support a DevOps continuous development and delivery cycle from idea to value realization.

DevSecOps	A mindset that "everyone is responsible for security" with the goal of safely distributing security decisions at speed and scale to those who hold the highest level of context without sacrificing the safety required.
Digital Transformation	The adoption of digital technology by a company to improve business processes, value for customers, and innovation.
Digital Value Stream	A value stream is anything that delivers a product or a service. A digital value stream is one that delivers a digital product or service.
Distributed Version Control System (DVCS)	The software revisions are stored in a distributed revision control system (DRCS), also known as a distributed version control system (DVCS).
DMZ (De-Militarized Zone)	A DMZ in network security parlance is a network zone in between the public internet and internal protected resources. Any application, server, or service (including APIs) that need to be exposed externally are typically placed in a DMZ. It is not uncommon to have multiple DMZs in parallel.
Dynamic Analysis	Dynamic analysis is the testing of an application by executing data in real-time with the objective of detecting defects while it is in operation, rather than by repeatedly examining the code offline.
Dynamic Application Security Testing (DAST)	Dynamic application security testing (DAST) is a process of testing an application or software product in an operating state.
EggPlant	Automated function and regression testing of enterprise applications. Licensed by Test Plant.
Elastic Infrastructure	Elasticity is a term typically used in cloud computing, to describe the ability of an IT infrastructure to quickly expand or cut back capacity and services without hindering or jeopardizing the infrastructure's stability, performance, security, governance, or compliance protocols.
eNPS	Employee Net Promoter Score (eNPS) is a way for organizations to measure employee loyalty. The Net Promoter Score, originally a customer service tool, was later used internally on employees instead of customers.
Entity Under Test (EUT)	This is a class of terms that refers to the names of types of entities that are being tested. These terms are often abbreviated to the form xUT where "x" represents a type of entity under test.
Epic	A collection of related user stories that may need to be worked on across multiple Sprints.

Erickson (Stages of Psychosocial Development)	Erik Erikson (1950, 1963) proposed a psychoanalytic theory of psychosocial development comprising eight stages from infancy to adulthood. During each stage, the person experiences a psychosocial crisis which could have a positive or negative outcome for personality development.
Error Budget	The error budget provides a clear, objective metric that determines how unreliable a service is allowed to be within a specific time period.
Error Budget Policies	An error budget policy enumerates the activity a team takes when they've exhausted their error budget for a particular service in a particular time period.
Error Tracking	Tools to easily discover and show the errors that the application may be generating, along with the associated data.
External Automation	Scripts and automation outside of a service that is intended to reduce toil.
Fail Early	A DevOps tenet referring to the preference to find critical problems as early as possible in a development and delivery pipeline.
Fail Often	A DevOps tenet which emphasizes a preference to find critical problems as fast as possible and therefore frequently.
Failure Rate	Fail verdicts per unit of time.
False Negative	A test incorrectly reports a verdict of "fail" when the EUT actually passed the purpose of the test.
False Positive	A test incorrectly reports a verdict of "pass" when the EUT actually failed the purpose of the test.
Feature Toggle	The practice of using software switches to hide or activate features. This enables continuous integration and testing a feature with selected stakeholders.
Federated Identity	A central identity used for access to a wide range of applications, systems, and services, but with a particular skew toward web-based applications. Also, often referenced as Identity-as-a-Service (IDaas). Any identity that can be reused across multiple sites, particularly via SAML or OAuth authentication mechanisms.

Fire Drills	A planned failure testing process focussed on the operation of live services including service failure testing as well as communication, documentation, and other human factor testing.
Flow	How people, products, or information move through a process. Flow is the first way of The Three Ways.
Flow of Value	A form of map that shows the end-to-end value stream. This view is usually not available within the enterprise.
Framework	The backbone for plugging in tools. Launches automated tasks, collects results from automated tasks.
Freedom and Responsibility	A core cultural value that with the freedom of self-management (such as afforded by DevOps) comes the responsibility to be diligent, to follow the advice process, and to take ownership of both successes and failures.
Frequency	How often an application is released.
Functional Testing	Tests to determine if the functional operation of the service is as expected.
Future State Map	A form of value stream map that helps you develop and communicate what the target end state should look like and how to tackle the necessary changes.
Fuzzing	Fuzzing or fuzz testing is an automated software testing practice that inputs invalid, unexpected, or random data into applications.
Gated Commits	Define and obtain consensus for the criterion of changes promoted between all CD pipeline stages such as Dev to CI stage / CI to packaging/delivery stage / Delivery to Deployment/Production stage.
Generative (DevOps) Culture	In a generative organization, alignment takes place through identification with the mission. The individual "buys into" what he or she is supposed to do and its effect on the outcome. Generative organizations tend to be proactive in getting the information to the right people by any means. necessary. (Westrum)
Generativity	A cultural view wherein long-term outcomes are of primary focus, which in turn drives investments and cooperation that enable an organization to achieve those outcomes.
Glass-Box	Same as Clear-Box Testing and White-Box Testing.

Goal-seeking tests	The purpose of the test is to determine an EUT's performance boundaries, using incrementally stresses until the EUT reaches peak performance. E.g. Determine the maximum throughput that can be handled without errors.
Golden Circle	A model by Simon Sinek that emphasizes an understanding of the business' "why" before focusing on the "what" and "how".
Golden Image	A template for a virtual machine (VM), virtual desktop, server, or hard disk drive. (TechTarget)
Goleman's Six Styles of Leadership	Daniel Goleman (2002) created the Six Leadership Styles and found, in his research, that leaders used one of these styles at any one time.
Governance, Risk Management and Compliance (GRC)	A team or software platform intended for concentrating governance, compliance, and risk management data, including policies, compliance requirements, vulnerability data, and sometimes asset inventory, business continuity plans, etc. In essence, a specialized document and data repository for security governance. Or a team of people who specialize in IT/security governance, risk management, and compliance activities. Most often non-technical business analyst resources.
Gray-Box	Test cases use a limited knowledge of the internal design structure of the EUT.
GUI testing	The purpose of the test is to determine if the graphical user interface operates as expected.
Guilds	A "community of interest" group that welcomes anyone and usually cuts across an entire organization. Similar to a Community of Practice.
Hand Offs	The procedure for transferring the responsibility of a particular task from one individual or team to another.
Hardening	Securing a server or infrastructure environment by removing or disabling unnecessary software, updating to known good versions of the operating system, restricting network-level access to only that which is needed, configuring logging in order to capture alerts, configuring appropriate access management, and installing appropriate security tools.
Helm Chart Registry	Helm charts are what describe related Kubernetes resources. Artifactory and Codefresh support a registry for maintaining master records of Helm Charts.
Heritage Reliability Engineer (HRE)	Applying the principles and practices of SRE to legacy applications and environments.

High-Trust Culture	Organizations with a high-trust culture encourage good information flow, cross-functional collaboration, shared responsibilities, learning from failures and new ideas.
Horizontal Scaling	Computing resources are scaled wider to increase the volume of processing. E.g. Add more computers and run more tasks in parallel.
Hypothesis-Backlog	A collection of requirements expressed as experiments.
Hypothesis-Driven Development (HDD)	A prototype methodology that allows product designers to develop, test, and rebuild a product until it's acceptable to the users.
Idempotent	CM tools (e.g., Puppet, Chef, Ansible, and Salt) claim that they are 'idempotent' by allowing the desired state of a server to be defined as code or declarations and automate steps necessary to consistently achieve the defined state time-after-time.
Identity	The unique name of a person, device, or the combination of both that is recognized by a digital system. Also referred to as an "account" or "user."
Identity and Access Management (IAM)	Policies, procedures, and tools for ensuring the right people have the right access to technology resources.
Identity as a Service (IDaaS)	Identity and access management services that are offered through the cloud or on a subscription basis.
Image-based test selection method	Build images are pre-assigned test cases. Tests cases are selected for a build by matching the image changes resulting from a build.
Immersive learning	A learning approach that guides teams with coaching and practice to help them learn to work in a new way.
Immutable	An immutable object is an object whose state cannot be modified after it is created. The antonym is a mutable object, which can be modified after it is created.
Immutable Infrastructures	Instead of instantiating an instance (server, container, etc.), with error-prone, time-consuming patches and upgrades (i.e. mutations), replace it with another instance to introduce changes or ensure proper behavior.
Impact-Driven Development (IDD)	A software development methodology that takes small steps towards achieving both impact and vision.

Implementation Under Test	The EUT is a software implementation. E.g. Embedded program is being tested.
Improvement Kata	A structured way to create a culture of continuous learning and improvement. (In Japanese business, Kata is the idea of doing things the "correct" way. An organization's culture can be characterized as its Kata through its consistent role modeling, teaching and coaching.)
Incentive model	A system designed to motivate people to complete tasks toward achieving objectives. The system may employ either positive or negative consequences for motivation.
Incident	Any unplanned interruption to an IT service or reduction in the quality of an IT service. Includes events that disrupt or could disrupt the service. (ITIL definition)
Incident Management	A process that restores normal service operation as quickly as possible to minimize business impact and ensure that agreed levels of service quality are maintained. (ITIL definition). Involves capturing the who, what, when of service incidents and the onward use of this data in ensuring service level objectives are being met.
Incident Response	An organized approach to addressing and managing the aftermath of a security breach or attack (also known as an incident). The goal is to handle the situation in a way that limits damage and reduces recovery time and costs.
Increment	Potentially shippable completed work that is the outcome of a Sprint.
Incremental Rollout	Deploying many small, gradual changes to a service instead of a few large changes. Users are incrementally moved across to the new version of the service until eventually all users are moved across. Sometimes referred to by colored environments e.g. Blue/green deployment.
Infrastructure	All of the hardware, software, networks, facilities, etc., required to develop, test, deliver, monitor and control or support IT services. The term IT infrastructure includes all of the information technology but not the associated people, processes, and documentation. (ITIL definition)
Infrastructure as Code (IaC)	The practice of using code (scripts) to configure and manage infrastructure.
Infrastructure Test	The purpose of the test is to verify the framework for EUT operating. E.g. verify specific operating system utilities function as expected in the target environment.

Infrastructure-as-a-Service (IaaS)	On-demand access to a shared pool of configurable computing resources.
Impact-Driven Development	A software development approach that takes small steps towards achieving both impact and vision.
Insights Driven	An insight-driven organization embeds analysis, data, and reasoning into the decision-making process, every day.
Integrated development environment (IDE)	An integrated development environment (IDE) is a software suite that consolidates the basic tools developers need to write and test software. Typically, an IDE contains a code editor, a compiler or interpreter, and a debugger that the developer accesses through a single graphical user interface (GUI). An IDE may be a standalone application, or it may be included as part of one or more existing and compatible applications. (TechTarget)
Integrated development environment (IDE) 'lint' checks	Linting is the process of running a program that will analyze code for potential errors (e.g., formatting discrepancies, non-adherence to coding standards and conventions, logical errors).
Internet of Things	A network of physical devices that connect to the internet and potentially to each other through web-based wireless services.
Internal Automation	Scripts and automation delivered as part of the service that is intended to reduce toil.
INVEST	A mnemonic was created by Bill Wake as a reminder of the characteristics of a quality user story.
ISO 31000	A family of standards that provide principles and generic guidelines on risk management.
Issue Management	A process for capturing, tracking, and resolving bugs and issues throughout the software development lifecycle.
IT Service Management (ITSM)	Adopting a process approach towards management, focusing on customer needs and IT services for customers rather than IT systems, and stressing continual improvement. (Wikipedia)
iTest	Tool licensed by Spirent Communications for creating automated test cases.

ITIL	Provides a best practices framework that organizations can adapt to deliver and maintain IT services to provide optimal value for all stakeholders, including the customer.
Jenkins	Jenkins is a freeware tool. It is the most popular master automation framework tool, especially for continuous integration task automation. Jenkins task automation centers around timed processes. Many test tools and other tools offer plugins to simplify integration with Jenkins.
Kaizen	The practice of continuous improvement.
Kanban	Method of work that pulls the flow of work through a process at a manageable pace.
Kanban Board	Tool that helps teams organize, visualize and manage work.
Karpman Drama Triangle	The drama triangle is a social model of human interaction. The triangle maps a type of destructive interaction that can occur between people in conflict.
Key Metrics	Something that is measured and reported upon to help manage a process, IT service or activity.
Key Performance Indicator (KPI)	Key performance indicators are the critical indicators of progress toward an intended result, providing a focus for improvement, and on what matters most.
Keywords-Based	Test cases are created using pre-defined names that reference programs useful for testing.
Knowledge Management	A process that ensures the right information is delivered to the right place or person at the right time to enable an informed decision.
Known Error	Problem with a documented root cause and a workaround. (ITIL definition)
Kolb's Learning Styles	David Kolb published his learning styles model in 1984; his experiential learning theory works on two levels: a four-stage cycle of learning and four separate learning styles.
Kotter's Dual Operating System	John Kotter describes the need for a dual operating system that combines the entrepreneurial capability of a network with the organizational efficiency of traditional hierarchy.

Kubernetes	Kubernetes is an open-source container-orchestration system for automating application deployment, scaling, and management. It was originally designed by Google and is now maintained by the Cloud Native Computing Foundation.
Kubler-Ross Change Curve	Describes and predicts the stages of personal and organizational reaction to major changes.
Lab-as-a-Service (LaaS)	Category of cloud computing services that provides a laboratory allowing customers to test applications without the complexity of building and maintaining the lab infrastructure.
Laloux (Culture Models)	Frederic Laloux created a model for understanding organizational culture.
Latency	Latency is the delay incurred in communicating a message, the time a message spends "on the wire" between the initial request being received e.g. by a server, and the response being received e.g. by a client.
Laws of Systems Thinking	In his book, 'The Fifth Discipline', Peter Senge outlines eleven laws that will help the understanding of business systems and to identify behaviors for addressing complex business problems.
Lean	Production philosophy that focuses on reducing waste and improving the flow of processes to improve overall customer value.
Lean (adjective)	Spare, economical. Lacking richness or abundance.
Lean Canvas	Lean Canvas is a 1-page business plan template.
Lean Enterprise	An organization that strategically applies the key ideas behind lean production across the enterprise.
Lean IT	Applying the key ideas behind lean production to the development and management of IT products and services.
Lean Manufacturing	Lean production philosophy derived mostly from the Toyota Production System.

Lean Product Development	Lean Product Development, or LPD, utilizes Lean principles to meet the challenges of Product Development.
Lean Startup	A system for developing a business or product in the most efficient way possible to reduce the risk of failure.
License Scanning	Tools, such as Blackduck and Synopsis, that check that licenses of your dependencies are compatible with your application, and approve or blacklist them.
Little's Law	A theorem by John Little that states that the long-term average number L of customers in a stationary system is equal to the long-term average effective arrival rate λ multiplied by the average time W that a customer spends in the system.
LoadRunner	A tool used to test applications, measuring system behavior, and performance under load. Licensed by HP.
Log	Serialized report of details such as test activities and EUT console logs.
Log Management	The collective processes and policies used to administer and facilitate the generation, transmission, analysis, storage, archiving, and ultimate disposal of the large volumes of log data created within an information system.
Logging	The capture, aggregation, and storage of all logs associated with system performance including, but not limited to, process calls, events, user data, responses, error, and status codes. Logstash and Nagios are popular examples.
Logic Bomb (Slag Code)	A string of malicious code used to cause harm to a system when the programmed conditions are met.
Longevity Test	The purpose of the test is to determine if a complete system performs as expected over an extended period of time
Machine Learning	Data analysis that uses algorithms that learn from data.
Malware	A program designed to gain access to computer systems, normally for the benefit of some third party, without the user’s permission
Many-factor Authentication	The practice of using at least 2 factors for authentication. The two factors can be of the same class.

Mean Time Between Deploys	Used to measure deployment frequency.
Mean Time Between Failures (MTBF)	The average time that a CI or IT service can perform its agreed function without interruption. Often used to measure reliability. Measured from when the CI or service starts working, until the time it fails (uptime). (ITIL definition)
Mean Time to Detect Defects (MTTD)	Average time required to detect a failed component or device.
Mean Time to Discovery	How long a vulnerability or software bug/defect exists before it's identified.
Mean Time to Patch	How long it takes to apply patches to environments once a vulnerability has been identified.
Mean Time to Repair/Recover (MTTR)	Average time required to repair/recover a failed component or device. MTTR does not include the time required to recover or restore service.
Mean Time to Restore Service (MTRS)	Used to measure time from when the CI or IT service fails until it is fully restored and delivering its normal functionality (downtime). Often used to measure maintainability. (ITIL definition).
Mental Models	A mental model is an explanation of someone's thought process about how something works in the real world.
Merge	The action of integrating software changes together into a software version management system.
Metric	Something that is measured and reported upon to help manage a process, IT service, or activity.
Metrics	This is a class of terms relevant to measurements used to monitor the health of a product or infrastructure.

Microprocess	A distinct activity that can be defined, designed, implemented, and managed independently and is generally associated with a primary service management practice. A microprocess may be integrated with other service management practices.
Microprocess Architecture	A collection of integrated microprocesses that collectively perform all of the activities necessary for an end-to-end service management practice to be successful.
Microservices	A software architecture that is composed of smaller modules that interact through APIs and can be updated without affecting the entire system.
Mindset	A person's usual attitude or mental state is their mindset.
Minimum Viable Process	The least amount needed in order for this process or microprocess to meet its Definition of Done.
Minimum Viable Product	Most minimal version of a product that can be released and still provide enough value that people are willing to use it.
Mock Object	Mock is a method/object that simulates the behavior of a real method/object in controlled ways. Mock objects are used in unit testing. Often a method under a test calls other external services or methods within it. These are called dependencies.
Model	Representation of a system, process, IT service, CI, etc. that is used to help understand or predict future behavior. In the context of processes, models represent pre-defined steps for handling specific types of transactions.
Model-Based	Test cases are automatically derived from a model of the entity under test. Example tool: Tricentis
Monitoring	The use of a hardware or software component to monitor the system resources and performance of a computer service.
Monitoring Tools	Tools that allow IT organizations to identify specific issues of specific releases and to understand the impact on end-users.
Monolithic	A software system is called "monolithic" if it has a monolithic architecture, in which functionally distinguishable aspects (for example data input and output, data processing, error handling, and the user interface) are all interwoven, rather than containing architecturally separate components.
Multi-factor Authentication	The practice of using 2 or more factors for authentication. Often used synonymously with 2-factor Authentication.

Multi-cloud	Multi-cloud DevOps solutions provide on-demand multi-tenant access to development and test environments.
Network Reliability Engineer (NRE)	Someone who applies a reliability engineering approach to measure and automate the reliability of networks.
Neuroplasticity	Describes the ability of the brain to form and reorganize synaptic connections, especially in response to learning or experience or following injury.
Neuroscience	The study of the brain and nervous system.
Non-functional requirements	Requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviors or functions (e.g., availability, reliability, maintainability, supportability); qualities of a system.
Non-functional tests	Defined as a type of service testing intending to check non-functional aspects such as performance, usability, and reliability of a software service.
Object Under Test (OUT)	The EUT is a software object or class of objects.
Observability	Observability is focused on externalizing as much data as you can about the whole service allowing us to infer what the current state of that service is.
Objectives and Key Results (OKRs)	Objectives and key results is a goal-setting framework used by individuals, teams, and organizations to define measurable goals and track their outcomes.
On-call	Being on-call means someone being available during a set period of time, and being ready to respond to production incidents during that time with appropriate urgency.
Open Source	Software that is distributed with its source code so that end-user organizations and vendors can modify it for their own purposes.
Operations (Ops)	Individuals involved in the daily operational activities needed to deploy and manage systems and services such as quality assurance analysts, release managers, system and network administrators, information security officers, IT operations specialists, and service desk analysts.
Operations Management	The function that performs the daily activities needed to deliver and support IT services and the supporting IT infrastructure at the agreed levels. (ITIL)

Ops	Individuals involved in the daily operational activities needed to deploy and manage systems and services such as quality assurance analysts, release managers, system and network administrators, information security officers, IT operations specialists, and service desk analysts.
Orchestration	An approach to building automation that interfaces or "orchestrates" multiple tools together to form a toolchain.
Organization Culture	A system of shared values, assumptions, beliefs, and norms that unite the members of an organization.
Organization Model	For DevOps, an approach that models Spotify's Squad approach for organizing IT.
Organizational Change	Efforts to adapt the behavior of humans within an organization to meet new structures, processes, or requirements.
OS Virtualization	A method for splitting a server into multiple partitions called "containers" or "virtual environments" in order to prevent applications from interfering with each other.
Outcome	Intended or actual results.
Outcome Mapping	A methodology for planning, monitoring, and evaluating development initiatives in order to bring about sustainable change.
Package Registry	A repository for software packages, artifacts, and their corresponding metadata. Can store files produced by an organization itself or for third-party binaries. Artifactory and Nexus are amongst the most popular.
Pages	Something for creating supporting web pages automatically as part of a CI/CD pipeline.
Patch	A software update designed to address (mitigate/remediate) a bug or weakness.
Patch management	The process of identifying and implementing patches.
Pathological Culture	Pathological cultures tend to view information as a personal resource, to be used in political power struggles (Westrum).
Penetration Testing	An authorized simulated attack on a computer system that looks for security weaknesses, potentially gaining access to the system's features and data.

People Changes	Focuses on changing attitudes, behaviors, skills, or performance of employees.
Performance Test	The purpose of the test is to determine an EUT meets its system performance criterion or to determine what a system's performance capabilities are.
Plan-Do-Check-Act	A four-stage cycle for process management and improvement attributed to W. Edwards Deming. Sometimes called the Deming Cycle or PDCA.
Platform-as-a-Service (PaaS)	Category of cloud computing services that provides a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the infrastructure.
Plugin	A pre-programmed integration between an orchestration tool and other tools. For example, many tools offer plugins to integrate with Jenkins.
Policies	Formal documents that define boundaries in terms of what the organization may or may not do as part of its operations.
Policy as Code	The notion that security principles and concepts can be articulated in code (e.g., software, configuration management, automation) to a sufficient degree that the need for an extensive traditional policy framework is greatly reduced. Standards and guidelines should be implemented in code and configuration, automatically enforced, and automatically reported on in terms of compliance, variance, or suspected violations.
Practice	A complete end-to-end capability for managing a specific aspect of service delivery (e.g. changes, incidents, service levels).
Practice Backlog	A prioritized list of everything that needs to be designed or improved for a practice including current and future requirements.
Practice/Microprocess Planning	A high-level event to define the goals, objectives, inputs, outcomes, activities, stakeholders, tools, and other aspects of a practice or microprocess. This meeting is not timeboxed.
Pre-Flight	This is a class of terms that refers to names of activities and processes that are conducted on an EUT prior to integration into the trunk branch.
Priority	The relative importance of an incident, problem, or change; based on impact and urgency. (ITIL definition)

Privileged Access Management (PAM)	Technologies that help organizations provide secured privileged access to critical assets and meet compliance requirements by securing, managing, and monitoring privileged accounts and access. (Gartner)
Problem	The underlying cause of one or more incidents. (ITIL definition)
Process	A structured set of activities designed to accomplish a specific objective. A process takes inputs and turns them into defined outputs. Related work activities that take specific inputs and produce specific outputs that are of value to a customer.
Process Changes	Focuses on changes to standard IT processes, such as software development practices, ITIL processes, change management, approvals, etc.
Process Owner	A role accountable for the overall quality of a process. It may be assigned to the same person who carries out the Process Manager role, but the two roles may be separate in larger organizations. (ITIL definition)
Process Standup	A time-boxed event of 15 minutes to inspect progress towards the Sprint Goal and identify impediments as quickly as possible.
Processing Time	The period during which one or more inputs are transformed into a finished product by a manufacturing or development procedure. (Business Dictionary)
Product Backlog	Prioritized list of functional and non-functional requirements for a system usually expressed as user stories.
Product Owner	An individual responsible for maximizing the value of a product and for managing the product backlog. Prioritizes, grooms, and owns the backlog. Gives the squad purpose.
Programming-Based	Test cases are created by writing code in a programming language. E.g. JavaScript, Python, TCL, Ruby
Project to Product	Changing ways of working from a large batch, waterfall project led approach, to a small batch, agile product (or value stream) approach.
Provision Platforms	Tools that provide platforms for provisioning infrastructure (e.g., Puppet, Chef, Salt).
Psychological Safety	Psychological safety is a shared belief that the team is safe for interpersonal risk-taking.

QTP	Quick Test Professional is a functional and regression test automation tool for software applications. Licensed by HP.
Quality Management	Tools that handle test case planning, test execution, defect tracking (often into backlogs), severity, and priority analysis. CA's Agile Central
Ranorex	GUI test automation framework for testing of desktop, web-based and mobile applications. Licensed by Ranorex.
Ransomware	Encrypts the files on a user's device or a network's storage devices. To restore access to the encrypted files, the user must pay a "ransom" to the cybercriminals, typically through a tough-to-trace electronic payment method such as Bitcoin.
RASP	Runtime Application Self-Protection
Regression testing	The purpose of the test is to determine if a new version of an EUT has broken some things that worked previously.
Regulatory compliance testing	The purpose of the test is to determine if an EUT conforms to specific regulatory requirements. E.g. verify an EUT satisfies government regulations for consumer credit card processing.
Release	Software that is built, tested, and deployed into the production environment.
Release Acceptance Criteria	Measurable attributes for a release package that determine whether a release candidate is acceptable for deployment to customers.
Release Candidate	A release package that has been prepared for deployment, may or may not have passed the Release.
Release Governance	Release Governance is all about the controls and automation (security, compliance, or otherwise) that ensure your releases are managed in an auditable and trackable way, in order to meet the need of the business to understand what is changing.
Release Management	The process that manages releases and underpins Continuous Delivery and the Deployment Pipeline.

Release Orchestration	Typically a deployment pipeline used to detect any changes that will lead to problems in production. Orchestrating other tools will identify performance, security, or usability issues. Tools like Jenkins and Gitlab CI can "orchestrate" releases.
Relevance	A Continuous Testing tenet which emphasizes a preference to focus on the most important tests and test results
Reliability	A measure of how long a service, component, or CI can perform its agreed function without interruption. Usually measured as MTBF or MTBSI. (ITIL definition)
Reliability Test	The purpose of the test is to determine if a complete system performs as expected under stressful and loaded conditions over an extended period of time.
Remediation	Action to resolve a problem found during DevOps processes. E.g. Roll-back changes for an EUT change that resulted in a CT test case fail verdict.
Remediation Plan	A plan that determines the actions to take after a failed change or release. (ITIL definition)
Request for Change (RFC)	Formal proposal to make a change. The term RFC is often misused to mean a change record, or the change itself. (ITIL definition)
Requirements Management	Tools that handle requirements definition, traceability, hierarchies & dependency. Often also handles code requirements and test cases for requirements.
Resilience	Building an environment or organization that is tolerant to change and incidents.
Response Time	Response time is the total time it takes from when a user makes a request until they receive a response.
REST	Representation State Transfer. The software architecture style of the worldwide web.
Restful API	Representational state transfer (REST) or RESTful services on a network, such as HTTP, provide scalable interoperability for requesting systems to quickly and reliably access and manipulate textual representations (XML, HTML, JSON) of resources using stateless operations (GET, POST, PUT, DELETE, etc.).

RESTful interface testing	The purpose of the test is to determine if an API satisfies its design criterion and the expectations of the REST architecture.
Return on Investment (ROI)	The difference between the benefit achieved and the cost to achieve that benefit, expressed as a percentage.
Review Apps	Allow code to be committed and launched in real-time – environments are spun up to allow developers to review their application.
Rework	The time and effort required to correct defects (waste).
Risk	A possible event that could cause harm or loss or affect an organization's ability to achieve its objectives. The management of risk consists of three activities: identifying risks, analyzing risks, and managing risks. The probable frequency and probable magnitude of future loss. Pertains to a possible event that could cause harm or loss or affect an organization's ability to execute or achieve its objectives.
Risk Event	A possible event that could cause harm or loss or affect an organization's ability to achieve its objectives. The management of risk consists of three activities: identifying risks, analyzing risks, and managing risks.
Risk Management Process	The process by which "risk" is contextualized, assessed and treated. From ISO 31000: 1) Establish context, 2) Assess risk, 3) Treat risk (remediate, reduce or accept).
Robot Framework	TDD framework created and supported by Google.
Role	Set of responsibilities, activities, and authorities granted to a person or team. A role is defined by a process. One person or team may have multiple roles. A set of permissions assigned to a user or group of users to allow a user to perform actions within a system or application.
Role-based Access Control (RBAC)	An approach to restricting system access to authorized users.
Roll-back	Software changes which have been integrated are removed from the integration.
Root Cause Analysis (RCA)	Actions take to identify the underlying cause of a problem or incident.

Rugged Development (DevOps)	Rugged Development (DevOps) is a method that includes security practices as early in the continuous delivery pipeline as possible to increase cybersecurity, speed, and quality of releases beyond what DevOps practices can yield alone.
Rugged DevOps	Rugged DevOps is a method that includes security practices as early in the continuous delivery pipeline as possible to increase cybersecurity, speed, and quality of releases beyond what DevOps practices can yield alone.
Runbooks	A collection of procedures necessary for the smooth operation of a service. Previously manual in nature they are now usually automated with tools like Ansible.
Runtime Application Self Protection (RASP)	Tools that actively monitor and block threats in the production environment before they can exploit vulnerabilities.
Sanity Test	A very basic set of tests that determine if a software is functional at all.
Scalability	Scalability is a characteristic of a service that describes its capability to cope and perform under an increased or expanding load.
Scaled Agile Framework (SAFE)	A proven, publicly available, framework for applying Lean-Agile principles and practices at an enterprise scale.
SCARF Model	A summary of important discoveries from neuroscience about the way people interact socially.
Scheduling	Scheduling: the process of planning to release changes into production.
Scrum	A simple framework for effective team collaboration on complex projects. Scrum provides a small set of rules that create "just enough" structure for teams to be able to focus their innovation on solving what might otherwise be an insurmountable challenge. (Scrum.org)
Scrum Pillars	Pillars that uphold the Scrum framework include Transparency, Inspection, and Adaption.
Scrum Team	A self-organizing, cross-functional team that uses the Scrum framework to deliver products iteratively and incrementally. The Scrum Team consists of a Product Owner, Developers, and a Scrum Master.
Scrum Values	A set of fundamental values and qualities underpinning the Scrum framework: commitment, focus, openness, respect and courage.

Scrum Master	An individual who provides process leadership for Scrum (i.e., ensures Scrum practices are understood and followed) and who supports the Scrum Team by removing impediments.
Secret Detection	Secret Detection aims to prevent that sensitive information, like passwords, authentication tokens, and private keys are unintentionally leaked as part of the repository content.
Secrets Management	Secrets management refers to the tools and methods for managing digital authentication credentials (secrets), including passwords, keys, APIs, and tokens for use in applications, services, privileged accounts, and other sensitive parts of the IT ecosystem.
Secure Automation	Secure automation removes the chance of human error (and wilful sabotage) by securing the tooling used across the delivery pipeline.
Security (Information Security)	Practices intended to protect the confidentiality, integrity, and availability of computer system data from those with malicious intentions.
Security as Code	Automating and building security into DevOps tools and practices, making it an essential part of toolchains and workflows.
Security tests	The purpose of the test is to determine if an EUT meets its security requirements. An example is a test that determines if an EUT processes login credentials properly.
Selenium	Popular open-source tool for software testing GUI and web applications.
Self-healing	Self-healing means the ability of services and underlying environments to detect and resolve problems automatically. It eliminates the need for manual human intervention.
Serverless	A code execution paradigm where no underlying infrastructure or dependencies are needed, moreover, a piece of code is executed by a service provider (typically cloud) who takes over the creation of the execution environment. Lambda functions in AWS and Azure Functions are examples.
Service	Enables the ability to do something when and how it is needed or desired. It enables its customers to achieve their objectives more efficiently and/or more effectively than they could without the service.
Service Desk	Single point of contact between the service provider and the users. Tools like Service Now are used for managing the lifecycle of services as well as internal and external stakeholder engagement.

Service Level Agreement (SLA)	Written agreement between an IT service provider and its customer(s) that defines key service targets and responsibilities of both parties. An SLA may cover multiple services or customers. (ITIL definition)
Service Level Indicator (SLI)	SLI's are used to communicate quantitative data about services, typically to measure how the service is performing against an SLO.
Service Level Objective (SLO)	An SLO is a goal for how well a product or service should operate. SLO's are set based on what an organization is expecting from a service.
Seven Pillars of DevOps	Seven distinct "pillars" provide a foundation for DevOps systems which include Collaborative Culture, Design for DevOps, Continuous Integration, Continuous Testing, Continuous Delivery and Deployment, Continuous Monitoring, and Elastic Infrastructure and Tools.
Shift Left	An approach that strives to build quality into the software development process by incorporating testing early and often. This notion extends to security architecture, hardening images, application security testing, and beyond.
SilkTest	Automated function and regression testing of enterprise applications. Licensed by Borland.
Simian Army	The Simian Army is a suite of failure-inducing tools designed by Netflix. The most famous example is Chaos Monkey which randomly terminates services in production as part of a Chaos Engineering approach.
Single Point of Failure (SPOF)	A single point of failure (SPOF) is a part of a system that, if it fails, will stop the entire system from working.
Site Reliability Engineering (SRE)	The discipline that incorporates aspects of software engineering and applies them to infrastructure and operations problems. The main goals are to create scalable and highly reliable software systems.
Smoke Test	A basic set of functional tests that are run immediately after a software component is built. Same as CI Regression Test.
Snapshot	Report of pass/fail results for a specific build.
Snippets	Stored and shared code snippets to allow collaboration around specific pieces of code. Also allows code snippets to be used in other code-bases. BitBucket and GitLab allow this.

SOAP	Simple Object Access Protocol (SOAP) is an XML-based messaging protocol for exchanging information among computers.
Software Composition Analysis	A tool that checks for libraries or functions in source code that have known vulnerabilities.
Software Defined Networking (SDN)	Software-Defined Networking (SDN) is a network architecture approach that enables the network to be intelligently and centrally controlled, or 'programmed,' using software applications.
Software Delivery Lifecycle (SDLC)	The process used to design, develop and test high quality software.
Software Version Management System	A repository tool which is used to manage software changes. Examples are: Azure DevOps, BitBucket, Git, GitHub, GitLab, VSTS.
Software-as-a-Service (SaaS)	Category of cloud computing services in which software is licensed on a subscription basis.
Source Code Tools	Repositories for controlling source code for key assets (application and infrastructure) as a single source of truth.
Spotify Squad Model	An organizational model that helps teams in large organizations behave like startups and be nimble.
Sprint	A period of 2-4 weeks during which an increment of product work is completed.
Sprint (Scrum)	A time-boxed iteration of work during which an increment of product functionality is implemented.
Sprint Backlog	Subset of the backlog that represents the work that must be completed to realize the Sprint Goal.
Sprint Goal	The purpose and objective of a Sprint, often expressed as a business problem that is going to be solved.

Sprint Planning	A 4 to 8-hour time-boxed event that defines the Sprint Goal, the increment of the Product Backlog that will be completed during the Sprint, and how it will be completed.
Sprint Retrospective	A 1.5 to 3-hour time-boxed event during which the Team reviews the last Sprint and identifies and prioritizes improvements for the next Sprint.
Sprint Review	A time-boxed event of 4 hours or less where the Team and stakeholders inspect the work resulting from the Sprint and update the Product Backlog.
Spyware	Software that is installed in a computer without the user's knowledge and transmits information about the user's computer activities over back to the threat agent.
Squads	A cross-functional, co-located, autonomous, self-directed team.
Stakeholder	Person who has an interest in an organization, project or IT service. Stakeholders may include customers, users and suppliers. (ITIL definition).
Stability	The sensitivity a service has to accept changes and the negative impact that may be caused by system changes. Services may have reliability, in that if functions over a long period of time, but may not be easy to change and so does not have stability.
Standard Change	Pre-approved, low risk change that follows a procedure or work instruction. (ITIL definition)
Static Application Security Testing (SAST)	A type of testing that checks source code for bugs and weaknesses.
Static Code Analysis	The purpose of the test is to detect source code logic errors and omissions such as memory leaks, unutilized variables, unutilized pointers.
Status Page	Service pages that easily communicate the status of services to customers and users.
Sticks	Negative incentives, for discouraging or punishing undesired behaviors.
Storage Security	A specialty area of security that is concerned with securing data storage systems and ecosystems and the data that resides on these systems.
Stormstack	A commercial orchestration tool based on event triggers instead of time-based.
StoStaKee	This stands for stop, start, and keep: this is an interactive time-boxed exercise focused on past events.

Strategic Sprint	A <4 week timeboxed Sprint during which strategic elements that were defined during Practice Planning are completed so that the Team can move on to designing the activities of the process.
Stream-Aligned Team	A team aligned to a single, valuable stream of work; this might be a single product or service, a single user story, or a single user persona.
Structural Changes	Changes in the hierarchy of authority, goals, structural characteristics, administrative procedures, and management systems.
Supplier	External (third party) supplier, manufacturer, or vendor responsible for supplying goods or services that are required to deliver IT services.
Synthetic Monitoring	Synthetic monitoring (also known as active monitoring, or semantic monitoring) runs a subset of an application's automated tests against the system on a regular basis. The results are pushed into the monitoring service, which triggers alerts in case of failures.
System of Record	A system of record is the authoritative data source for a data element or data entity.
System Test	The purpose of the test is to determine if a complete system performs as expected in its intended configurations.
System Under Test (SUT)	The EUT is an entire system. E.g. Bank teller machine is being tested.
Tag-Based Test Selection Method	Tests and Code modules are pre-assigned tags. Tests are selected for a build matching pre-assigned tags.
Target Operating Model	A description of the desired state of the operating model of an organization.
Teal Organization	An emerging organizational paradigm that advocates a level of consciousness including all previous world views within the operations of an organization.
Team Dynamics	A measurement of how a team works together. Includes team culture, communication styles, decision-making ability, trust between members, and the willingness of the team to change.
Team Topologies	An approach to organizing business and technology teams for fast flow, providing a practical, step-by-step, adaptive model for organizational design and team interaction.

Techno-Economic Paradigm Shifts	Techno-economic paradigm shifts are at the core of the general, innovation-based theory of economic and societal development as conceived by Carlota Perez.
Telemetry	Telemetry is the collection of measurements or other data at remote or inaccessible points and their automatic transmission to receiving equipment for monitoring.
Test Architect	Person who has responsibility for defining the overall end-to-end test strategy for an EUT.
Test Artifact Repository	Database of files used for testing.
Test Campaign	A test campaign may include one or more test sessions.
Test Case	Set of test steps together with data and configuration information. A test case has a specific purpose to test at least one attribute of the EUT.
Test Creation Methods	This is a class of test terms that refers to the methodology used to create test cases.
Test-Driven Development (TDD)	Test-driven development (TDD) is a software development process in which the developer writes a test before composing code. They then follow this process: 1. Write the test 2. Run the test and any others that are relevant and see them fail 3. Write the code 4. Run test(s) 5. Refactor code if needed 6. Repeat Unit level tests and/or application tests are created ahead of the code that is to be tested.
Test Duration	The time it takes to run a test. E.g. # hours per test

Test Environment	The test environment refers to the operating system (e.g. Linus, windows version, etc.), the configuration of software (e.g. parameter options), dynamic conditions (e.g. CPU and memory utilization), and physical environment (e.g. power, cooling) in which the tests are performed.
Test Fast	A CT tenet referring to accelerated testing.
Test Framework	A set of processes, procedures, abstract concept,s and environments in which automated tests are designed and implemented.
Test Harness	A tool which enables the automation of tests. It refers to the system test drivers and other supporting tools that requires to execute tests. It provides stubs and drivers which are small programs that interact with the software under test.
Test Hierarchy	This is a class of terms describes the organization of tests into groups.
Test Methodology	This class of terms identifies the general methodology used by a test. Examples are White Box, Black Box
Test result repository	Database of test results.
Test Results Trend-based	A matrix of correlation factors correlates test cases and code modules according to test results (verdict).
Test Roles	This class of terms identifies general roles and responsibilities for people relevant to testing.
Test Script	Automated test case. A single test script may be implemented with one or more test cases depending on the data.
Test Selection Method	This class of terms refers to the method used to select tests to be executed on a version of an EUT.

Test Session	Set of one or more test suites that are run together on a single build at a specific time.
Test Suite	Set of test cases that are run together on a single build at a specific time.
Test Trend	History of verdicts.
Test Type	The class which indicates the purpose of the test.
Test Version	The version of files used to test a specific build.
Tester	An individual who has the responsibility to test a system or service.
Testing Tools	Tools that verify code quality before passing the build.
The Advice Process	Any person deciding must seek advice from everyone meaningfully affected by the decision and people with expertise in the matter. Advice received must be taken into consideration, though it does not have to be accepted or followed. The objective of the advice process is not to form a consensus, but to inform the decision-maker so that they can make the best decision possible. Failure to follow the advice process undermines trust and unnecessarily introduces risk to the business.
The Checkbox Trap	The situation wherein an audit-centric perspective focuses exclusively on "checking the box" on compliance requirements without consideration for overall security objectives.
The Power of TED	The Power of TED* offers an alternative to the Karpman Drama Triangle with its roles of Victim, Persecutor, and Rescuer. The Empowerment Dynamic (TED) provides the antidote roles of Creator, Challenger, and Coach and a more positive approach to life's challenges.
The Sprint	A period of <4 weeks during which an increment of work is completed.

The Three Pillars of Empiricism	Three pillars uphold every implementation of empirical process control: transparency, inspection, and adaptation.
The Three Ways	Key principles of DevOps – Flow, Feedback, Continuous experimentation, and learning.
Theory of Constraints	Methodology for identifying the most important limiting factor (i.e., constraint) that stands in the way of achieving a goal and then systematically improving that constraint until it is no longer the limiting factor.
Thomas Kilmann Inventory (TKI)	Measures a person's behavioral choices under certain conflict situations.
Threat Agent	An actor, human or automated, that acts against a system with intent to harm or compromise that system. Sometimes also called a "Threat Actor."
Threat Detection	Refers to the ability to detect, report, and support the ability to respond to attacks. Intrusion detection systems and denial-of-service systems allow for some level of threat detection and prevention.
Threat Intelligence	Information pertaining to the nature of a threat or the actions a threat may be known to be perpetrating. May also include "indicators of compromise" related to a given threat's actions, as well as a "course of action" describing how to remediate the given threat action.
Threat Modeling	A method that ranks and models potential threats so that the risk can be understood and mitigated in the context of the value of the application(s) to which they pertain.
Time to Insight Actioned	The time between having an idea, delivering it to the customer, learning and actioning the insight from that learning.
Time to Learning	The time between conceiving an idea and learning how it was received based on customer feedback.
Time to Market	The period of time between when an idea is conceived and when it is available to customers.
Time to Value	The measure of the time it takes for the business to realize value from a feature or service.
Time Tracking	Tools that allow for time to be tracked, either against individual issues or other work or project types.

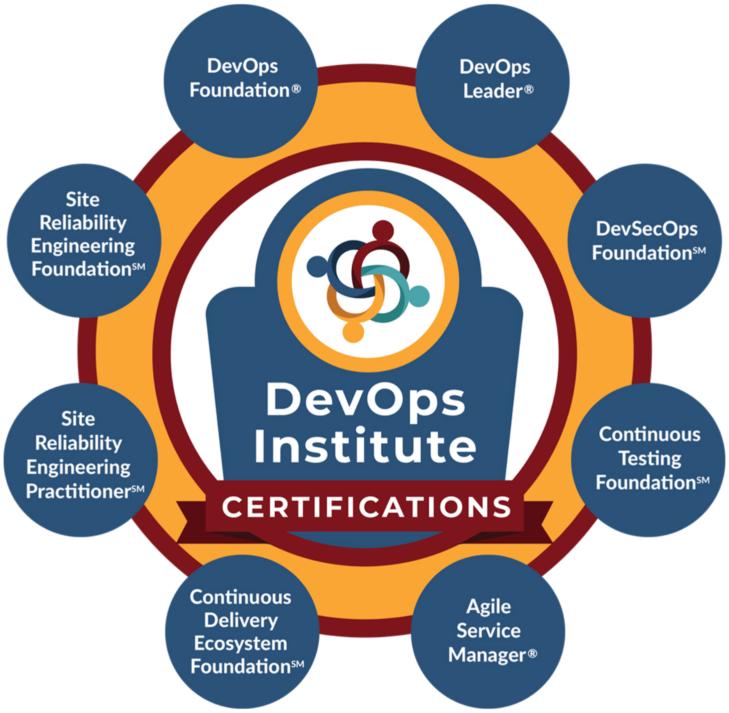
Timebox	The maximum duration of a Scrum event.
Toil	A kind of work tied to running a production service that tends to be manual, repetitive, automatable, tactical, devoid of enduring value.
Tool	This class describes tools that orchestrate, automate, simulate and monitor EUT's and infrastructures.
Toolchain	A philosophy that involves using an integrated set of complimentary task-specific tools to automate an end-to-end process (vs. a single-vendor solution).
Touch Time	In a Lean Production system the touch time is the time that the product is actually being worked on, and value is being added.
Tracing	Tracing provides insight into the performance and health of a deployed application, tracking each function or microservice which handles a given request.
Traffic Volume	The amount of data sent and received by visitors to a service (e.g. a website or API).
Training From the Back of the Room	An accelerated learning model in line with agile values and principles using the 4Cs instructional design "map" (Connection, Concept, Concrete Practice, Conclusion).
Transformational Leadership	A leadership model in which leaders inspire and motivate followers to achieve higher performance by appealing to their values and sense of purpose, facilitating wide-scale organizational change (State of DevOps Report, 2017).
Tribe Lead	A senior technical leader that has broad and deep technical expertise across all the squads' technical areas. A group of squads working together on a common feature set, product, or service is a tribe in Spotify's definitions.
Tribes	A collection of squads with a long-term mission that work on/in a related business capability.
Trojan (horses)	Malware that carries out malicious operations under the appearance of a desired operation such as playing an online game. A Trojan horse differs from a virus because the Trojan binds itself to non-executable files, such as image files, audio files whereas a virus requires an executable file to operate.
Trunk	The primary source code integration repository for a software product.

Unit Test	The purpose of the test is to verify code logic.
Usability Test	The purpose of the test is to determine if humans have a satisfactory experience when using an EUT.
User	Consumer of IT services. Or, the identity asserted during authentication (aka username).
User and Entity Behavior Analytics (UEBA)	A machine learning technique to analyze normal and “abnormal” user behavior with the aim of preventing the latter.
User Story	A brief statement used to describe a requirement from a user’s perspective. User stories are used to facilitate communication, planning, and negotiation activities between the stakeholders and the Agile Service Management Team.
Value Added Time	The amount of time spent on an activity that creates value (e.g., development, testing).
Value Cycle	The lifecycle stages of the value stream from ideation to value realization.
Value Efficiency	Being able to produce value with the minimum amount of time and resources.
Value Stream	All of the activities needed to go from a customer request to a delivered product or service.
Value Stream Map	Visually depicts the end-to-end flow of activities from the initial request to value creation for the customer.
Value Stream Mapping	A lean tool that depicts the flow of information, materials, and work across functional silos with an emphasis on quantifying waste, including time and quality.
Value Stream Management	Value Stream Management is a combination of people, processes, and technology that maps, optimizes, visualizes, measures, and governs business value flow through heterogeneous software delivery pipelines from idea through development and into production.
Value Stream Management Platform	Software that manages value streams.

Variable Speed IT	An approach where traditional and digital processes co-exist within an organization while moving at their own speed.
Velocity	The measure of the quantity of work done in a pre-defined interval. The amount of work an individual or team can complete in a given amount of time.
Verdict	Test result classified as Fail, Pass, or Inconclusive.
Version control tools	Ensure a 'single source of truth' and enable change control and tracking for all production artifacts.
Vertical Scaling	Computing resources are scaled higher to increase processing speed e.g. using faster computers to run more tasks faster.
Virus (Computer)	Malicious executable code attached to a file that spreads when an infected file is passed from system to system that could be harmless (but annoying) or it could modify or delete data.
Voice of the Customer (VOC)	A process that captures and analyzes customer requirements and feedback to understand what the customer wants.
Vulnerability	A weakness in a design, system, or application that can be exploited by an attacker.
Vulnerability Intelligence	Information describing a known vulnerability, including affected software by version, the relative severity of the vulnerability (for example, does it result in an escalation of privileges for a user role, or does it cause a denial of service), the exploitability of the vulnerability (how easy/hard it is to exploit), and sometimes current rate of exploitation in the wild (is it being actively exploited or is it just theoretical). This information will also often include guidance on what software versions are known to have remediated the described vulnerability.
Vulnerability management	The process of identifying and remediating vulnerabilities.
Wait Time	The amount of time wasted on waiting for work (e.g., waiting for development and test infrastructure, waiting for resources, waiting for management approval).
Waste (Lean Manufacturing)	Any activity that does not add value to a process, product or service.

Water-scrum-fall	A hybrid approach to application lifecycle management that combines waterfall and Scrum development can complete in a given amount of time.
Waterfall (Project Management)	A linear and sequential approach to managing software design and development projects in which progress is seen as flowing steadily (and sequentially) downwards (like a waterfall).
Weakness	An error in software that can be exploited by an attacker to compromise the application, system, or the data contained therein. Also called a vulnerability.
Web Application Firewall (WAF)	Tools that examine traffic being sent to an application and can block anything that looks malicious.
Web IDE	Tools that have a web client integrated development environment. Enables developer productivity without having to use a local development tool.
Westrum (Organization Types)	Ron Westrum developed a typology of organizational cultures that includes three types of organizations: Pathological (power-oriented), Bureaucratic (rule-oriented) and Generative (performance-oriented).
White-Box Testing (or Clear-, Glass-, Transparent-Box Testing or Structural Testing)	Test cases use extensive knowledge of the internal design structure or workings of an application, as opposed to its functionality (i.e. Black-Box Testing).
Whitelisting	Application whitelisting is the practice of specifying an index of approved software applications that are permitted to be present and active on a computer system.
Wicked Questions	Wicked questions are used to expose the assumptions which shape our actions and choices. They are questions that articulate the embedded, and often contradictory assumptions, we hold about an issue, a problem or a context.
Wiki	Knowledge sharing can be enabled by using tools like Confluence which create a rich Wiki of content
Wilber's Quadrants	A model that recognises four modes of general approach for human beings. Two axes are used: on one axis people tend towards individuality OR collectivity.
Work in Progress (WIP)	Any work that has been started but has not been completed.

Workaround	A temporary way to reduce or eliminate the impact of incidents or problems. May be logged as a known error in the Known Error Database. (ITIL definition).
World Café	Is a structured conversational process for knowledge sharing in which groups of people discuss a topic at several tables, with individuals switching tables periodically and getting introduced to the previous discussion at their new table by a "table host".
Worms (Computer)	Worms replicate themselves on a system by attaching themselves to different files and looking for pathways between computers. They usually slow down networks and can run by themselves (where viruses need a host program to run).


DevOps Institute
CERTIFICATIONS
DevOps Foundation®
DevOps Leader®
DevSecOps Foundation℠
Continuous Testing Foundation℠
Agile Service Manager®
Continuous Delivery Ecosystem Foundation℠
Site Reliability Engineering Practitioner℠
Site Reliability Engineering Foundation℠

Your Path to DevOps Success

DevOps Institute is dedicated to advancing the human elements of DevOps success. Our goal is to help advance careers and support emerging practices using a role-based approach to certification which focuses on the most modern competencies and hireable skills required by today's organizations adopting DevOps.

Take the next steps in your learning and certification journey to DevOps success.

Click on a certification or visit
www.devopsinstitute.com/certifications
to learn more.

Become a Member

Join the fastest growing global community of DevOps practitioners and professionals and gain access to invaluable learning content, the latest news, events, emerging practices, develop your network and advance your career.

You belong.

(devopsinstitute.com/membership)